FP/EDF, a non-preemptive scheduling combiningfixed priorities and deadlines:uniprocessor and distributed cases

Steven Martin, Pascale Minet, Laurent George

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

FP/EDF, a non-preemptive scheduling combining fixed priorities and deadlines: uniprocessor and distributed cases

Steven MARTIN — Pascale MINET — Laurent GEORGE

N° 5112 – version 2

version initiale February 2004 – version révisée April 2004

Thème COM

Rapport de recherche

ISSN 0249-6399

FP/EDF, a non-preemptive scheduling combining fixed priorities and deadlines: uniprocessor and distributed cases

Steven MARTIN* , Pascale MINET† , Laurent GEORGE‡

Thème COM — Systèmes communicants
Projet HIPERCOM

Rapport de recherche n° 5112 – version 2§— version initiale February 2004 — version révisée April 2004 39 pages

Abstract: In this paper, we focus on a non-preemptive scheduling of sporadic flows, combining fixed priorities and deadlines. This scheduling is called FP/EDF. With any flow are associated a fixed priority denoting the importance of the flow and a delivery deadline. A packet m can be transmitted only if there is no waiting packet with a fixed priority higher than m and no waiting packet with the same fixed priority as m but with a smaller deadline. We are interested in the worst case response time of flows, both in uniprocessor and distributed cases. In the uniprocessor case, we prove that any sporadic flow set feasible with the classical Fixed Priority scheduling is feasible with FP/EDF. The converse is false, as shown by an example. Moreover, we show that when all flows sharing the same fixed priority have the same processing time, any sporadic flow set feasible with FP/FIFO is feasible with FP/EDF, but the converse is false. We then establish new results with FP/EDF in a distributed context, when all flows follow the same sequence of nodes. The absolute deadline of a packet that is considered by any scheduler is computed on the first node visited and then left unchanged by any other node. We show in such conditions how to compute an upper bound on the end-to-end response time of any flow. For this, we use a worst case analysis based on the trajectory approach. Results obtained for some configurations are exact. In all configurations, these results are compared with those provided by the classical holistic approach. We show that our results are largely better.

* Ecole Centrale d'Electronique, LACCSC, 53 rue de Grenelle, 75007 Paris, *steven.martin@ece.fr*
† INRIA Rocquencourt, 78153 Le Chesnay Cedex, *pascale.minet@inria.fr*
‡ Université Paris 12, LIIA, 120 rue Paul Armangot, 94400 Vitry, *george@univ-paris12.fr*
§ Dans cette version, les preuves ont été améliorées et des exemples ont été ajoutés.

Key-words: Fixed priority scheduling, class-based scheduling, EDF, FIFO, worst case response time, end-to-end response time, deterministic guarantee, quality of service, trajectory approach, holistic approach.

FP/EDF, un ordonnancement non-préemptif à base de priorités fixes et d'échéances : cas uniprocesseur et distribué

Résumé : Dans ce rapport, nous nous intéressons à un ordonnancement non-préemptif de flux sporadiques combinant priorités fixes et échéances. Cet ordonnancement est appelé FP/EDF. A chaque flux sont associées une priorité fixe dénotant l'importance du flux et une échéance de remise. Un paquet m ne peut être transmis que s'il n'existe aucun paquet en attente avec une priorité fixe supérieure à celle de m et aucun paquet en attente de priorité fixe égale à celle de m mais avec une échéance plus petite. Nous montrons comment calculer le temps de réponse pire cas de flux en mono-processeur puis en environnement distribué. En mono-processeur, nous prouvons que tout jeu de flux sporadique faisable avec les priorités fixes classiques est faisable avec FP/EDF. La réciproque est fausse, comme le montre l'exemple présenté. De plus, nous montrons que lorsque les flux de même priorité fixe ont le même temps de traitement, tout jeu de flux sporadique faisable avec FP/FIFO est faisable avec FP/EDF, mais la réciproque est fausse. Nous établissons ensuite de nouveaux résultats avec FP/EDF dans un environnement distribué où tous les flux visitent la même séquence de noeuds. L'échéance absolue d'un paquet qui est considérée par tout ordonnanceur est calculée sur le premier noeud visité et demeure inchangée sur tout noeud visité. Nous montrons dans ces conditions comment calculer un majorant du temps de réponse de bout-en-bout pour chaque flux. Pour cela, nous avons recours à une analyse pire cas basée sur l'approche par trajectoire. Les résultats obtenus pour certaines configurations sont exacts. Dans toutes les configurations, ces résultats sont comparés avec ceux fournis par la classique approche holistique. Nous montrons que nos résultats sont bien meilleurs.

Mots-clés : Ordonnancement par priorité fixe, ordonnancement par classe, EDF, FIFO, temps de réponse pire cas, temps de réponse de bout-en-bout, garantie déterministe, qualité de service, approche par trajectoire, approche holistique.

Contents

1 Introduction

Fixed Priority scheduling has been extensively studied in the last years [1, 2]. It exhibits interesting properties:

- The impact of a flow τ_i is limited to flows having priorities smaller than this of τ_i.
- It is easy to implement.
- It is well adapted for service differentiation: flows with high priorities have smaller response times.
- It is well adapted to account for the packet importance from the application point of view.

In this paper, we focus on non-preemptive Fixed Priority scheduling, denoted FP, considering the local scheduling of flows sent on a network. Indeed, with regard to flow scheduling, the assumption generally admitted is that packet transmission is not preemptive. Moreover, several flows may have to share the same priority in the following cases:

- when the number of priorities available on a processor is less than the number of flows considered;
- when the priority of a flow is determined by external constraints and cannot be chosen arbitrarily;
- when a class-based scheduling is used: all flows within a class having the same Fixed Priority.

The classical analysis of Fixed Priority scheduling assumes that flows sharing the same priority are scheduled arbitrarily. The worst case response times obtained with this analysis can be improved, if the scheduling algorithm of flows sharing the same priority is accounted for. We propose to schedule such flows according to EDF (Earliest Deadline First). Indeed, EDF has been proved optimal [2, 3] for a uniprocessor in both preemptive and non-preemptive context when the packets release times are not known in advance. EDF optimality means that if for a given scheduling problem, EDF fails to find a feasible solution, there is no solution for this problem. That is why, we propose a new scheduling algorithm, called FP/EDF, combining fixed priorities and deadlines. The deadline is the second arbitration criterion used to schedule flow packets.

In this paper, we first determine the worst case response time of a flow in the uniprocessor case. Our solution enables to improve the worst case response times of flows. Moreover, any flow set feasible with Fixed Priority scheduling is feasible with FP/EDF, but the converse is false. Hence the interest of using FP/EDF.

Then, we focus on the distributed case, when all flows follow the same sequence of nodes (the same *line*). We show how to compute a bound on the end-to-end response time of any flow, using the trajecttory approach.

Our results on the worst case end-to-end response time obtained with FP/EDF scheduling can be used in various configurations:

- In a *Differentiated Services* architecture [5], several classes are defined, each having its own fixed priority. The highest priority class, that is the *Expedited Forwarding* (EF) class, is scheduled Fixed Priority with the other classes. Moreover, if packets belonging to the EF class need to be differentiated, different deadlines can be assigned to these packets. Hence, FP/EDF scheduling can be used to provide the requested differentiation.
- In an *Integrated Services* architecture [6], a fixed priority and a deadline are assigned to each flow. FP/EDF scheduling is used to provide shorter response times to high priority flows and to flows with the same priority but with short deadlines.
- In an *hybrid* architecture, some flows are managed per class, whereas others are managed individually.

In the three configurations mentionned, several flows can share the same fixed priority. Hence the interest of FP/EDF.

The rest of the paper is organized as follows. In Section 2, we focus on FP/EDF in a uniprocessor context and compute the worst case respone time of any flow. In Section 3, we compare our results with those provided by the classical fixed priority scheduling and show the benefit of FP/EDF. We show that these results can also be applied to FP/FIFO. Moreover, we prove that any sporadic flow set such that all flows sharing the same priority have the same processing time, feasible with FP/FIFO is feasible with FP/EDF. The converse is false. We then focus on the distributed case. Section 4 briefly discusses related work in the computation of worst case end-to-end response time. In Section 5, we show how to compute an upper bound on the end-to-end response time of any flow, based on a worst case analysis, when all flows follow the same sequence of nodes (the same *line*). Then, in Section 6, we compare our results, obtained by applying the trajectory approach, with the exact worst case end-to-end response times and with the results provided by the holistic approach. The exact values are obtained by a simulation tool we have designed. This simulation tool does an exhaustive analysis. Finally, we conclude the paper in Section 7.

Throughout this paper, we assume that time is discrete. [7] shows that results obtained with a discrete scheduling are as general as those obtained with a continuous scheduling when all flow parameters are multiples of the node clock tick. In such conditions, any set of flows is feasible with a discrete scheduling if and only if it is feasible with a continuous scheduling.

2 FP/EDF scheduling in a uniprocessor context

We first recall in Subsection 2.1 some notations and classical results used in hard real-time scheduling. Then, we determine in Subsection 2.2, for a given flow τ_i, the worst case scenario leading to the worst case response time of τ_i. In Subsection 2.3, we compute the latest starting time of a τ_i packet. Finally, in Subsection 2.4, we show how to compute the worst case response time of τ_i.

2.1 Notations and classical results

We consider a set $\tau = \{\tau_1, ..., \tau_n\}$ of n sporadic flows. Each flow τ_i is defined by:

- T_i, the minimum interarrival time (called period) between two successive packets of τ_i;
- C_i, the maximum processing time of any packet of τ_i;
- J_i, the maximum release jitter of packets of τ_i;
- D_i, the delivery deadline of any packet of τ_i.

This characterization is well adapted to real-time flows (e.g. process control, voice and video, sensor and actuator). Due to the scheduling model, any flow τ_i has a fixed priority P_i. We then denote:

- $gp(i) = \{j \in [1, n]$, such that $P_j > P_i\}$;
- $sp(i) = \{j \in [1, n], j \neq i$, such that $P_j = P_i\}$;
- $lp(i) = \{j \in [1, n]$, such that $P_j < P_i\}$;
- $hp_t(i) = \{j \in sp(i)$, such that $D_j - J_j \leq t + D_i\}$;
- $\overline{hp}_t(i) = \{j \in sp(i)$, such that $D_j - J_j > t + D_i\}$.

As said in the introduction, packet scheduling is non-preemptive. Hence, despite the high priority of any packet m, released at time t, a packet with a lower priority can delay m processing due to non-preemption. Indeed, if a packet m of any flow τ_i is released while a packet m' belonging to $lp(i) \cup \overline{hp}_t(i)$ is being processed, m has to wait until m' completion.

It is important to notice that the non-preemptive effect is not limited to this waiting time. The delay incurred by packet m directly due to m' may lead to consider packets belonging to $gp(i) \cup hp_t(i)$, released after m but before m starts its execution. The following lemma establishes the maximum delay incurred by a packet directly due to another packet belonging to $lp(i) \cup \overline{hp}_t(i)$.

Lemma 1 *In the uniprocessor case, the maximum delay incurred by any packet of any flow τ_i, released at time t, directly due to packets belonging to $lp(i) \cup \overline{hp}_t(i)$ is equal to:*

$\max\left(0\,;\, C_{\overline{max}_{i,t}} - 1\right)$, *where:*

$$C_{\overline{max}_{i,t}} = \begin{array}{l} \max_{j\,\in\, lp(i)\cup\overline{hp}_t(i)}\{C_j\} \textit{ if } lp(i)\cup\overline{hp}_t(i) \neq \emptyset, \\ 0 \textit{ otherwise.} \end{array}$$

Proof: Let us consider the busy period of level P_i where the packet m of flow τ_i is processed. Let time 0 be the beginning of the busy period and time t be the request time of m in this busy period. Any packet $m' \in lp(i) \cup \overline{hp}_t(i)$ can not delay m if released at a time greater than or equal to 0, as all packets scheduled after time 0 in this busy period have a priority higher than or equal to P_i. Hence, packet m' can delay m only if released before time 0. The maximum delay is obtained when m' is the packet of maximum execution duration $\in lp(i) \cup \overline{hp}_t(i)$ released at time -1. ■

We now recall some definitions about busy periods.

Definition 1 *An idle time t is a time such that all packets arrived before t have been processed at time t.*

Definition 2 *A busy period is defined by an interval $[t, t')$ such that t and t' are both idle times and there is no idle time $\in (t, t')$.*

Definition 3 *An idle time t of level P_i is a time such that all packets with a priority greater than or equal to P_i and arrived before t have been processed at time t.*

Definition 4 *A level P_i busy period is defined by an interval $[t, t')$ such that t and t' are both idle times of level P_i and there is no idle time of level $P_i \in (t, t')$.*

Let $W_i(t)$ denote the latest starting time of the packet of τ_i requested at time t and R_i denote the worst case response time of flow τ_i. A packet m of flow τ_i, requested at time t, is delayed by:

- packets of flows having a priority strictly greater than P_i and arrived at a time $\leq W_i(t)$;
- packets of flows having a priority equal to P_i and a deadline less than $t + D_i$, arrived at a time $\leq W_i(t)$;
- a packet of a flow in $lp(i) \cup \overline{hp}_t(i)$;

Definition 5 *For any flow τ_i, the processor utilization factor for the flows belonging to $gp(i)$, denoted $U_{gp(i)}$, is the fraction of processor time spent to process packets belonging to $gp(i)$. It is equal to $\sum_{j \in gp(i)}(C_j/T_j)$.*

We determine in Subsection 2.2 the latest starting time of any packet m of flow τ_i. This time can be computed as a limit of a series, that we study in Subsection 2.3. Thanks to this analysis, we give in Subsection 2.4 an upper bound on the worst case response time of flow τ_i.

2.2 Latest starting time computation

We first determine the scenarios that lead to the worst case response time of flow τ_i. Notice that any packet of flow τ_j requested before time 0, the beginning of a busy period is subject to a release jitter such that this packet is released at time 0. The workload generated by τ_j is maximum when the first packet of τ_j is requested at time $-J_j$ and all packets are periodic with period T_j.

Lemma 2 *In the uniprocessor case, when all flows are scheduled according to* FP/EDF*, the worst case response time of the packet of flow τ_i, requested at time t, is reached in the first busy period of a scenario where:*

- *$\forall$ flow $\tau_j \in gp(i) \cup hp_t(i)$, the first packet of τ_j is requested at time $-J_j$ and all other packets of flow τ_j are periodic, with period T_j;*
- *a packet of the flow τ_j belonging to $lp(i) \cup \overline{hp}_t(i)$, if any, with the maximum processing time is released at time -1;*
- *the first packet of flow τ_i is requested at time t_i^0, with $-J_i \leq t_i^0 < -J_i + T_i$ and $\exists k \in \mathbb{N}$ such that $t = t_i^0 + k \cdot T_i$ and all other packets of τ_i are periodic, with period T_i.*

Proof: Let us consider the first busy period of a scenario S in which a packet m of flow τ_i, requested at time t, is processed. If m is requested before time 0, it is released at time 0, due to the release jitter. The processing of packet m can be delayed by:

1. packets belonging to flow $\tau_j \in gp(i)$, requested before m starts its execution.
2. packets belonging to flow $\tau_j \in hp_t(i)$: packets having a priority P_i with a deadline D_j such that $D_j - J_j \leq t + D_i$, these packets must have been requested before m starts its processing.
3. packets belonging to flow τ_i, requested before m.
4. a packet belonging to a flow $\tau_j \in lp(i) \cup \overline{hp}_t(i)$.

We now modify scenario S to worsen the response time of packet m. The workload generated by packets of the first and second category is maximized when all flows $\tau_j \in gp(i) \cup hp_t(i)$ are first requested at time $-J_j$ and then periodically, with period T_j. In the same way, the workload generated by packets of the third category is maximized when the first packet of flow τ_i is requested at time t_i^0, with $-J_i \leq t_i^0 < -J_i + T_i$ and $\exists\, k \in \mathbb{N}$ such that $t = t_i^0 + k \cdot T_i$, and all other packets of τ_i are periodic, with period T_i. Finally, to maximize the delay due to a packet belonging to $lp(i) \cup \overline{hp}_t(i)$, a packet of the flow $\tau_j \in lp(i) \cup \overline{hp}_t(i)$, if any, with the maximum processing time is released at time -1. In this new scenario, the response time of packet m is either left unchanged or worse. ■

Lemma 3 *In the uniprocessor case, when all flows are scheduled according to* FP/EDF*, then for any packet of any flow τ_i, requested at time t, its latest starting time is given by:*

$$W_i(t) = \sum_{j \in gp(i)} \left(1 + \left\lfloor \frac{W_i(t)+J_j}{T_j} \right\rfloor\right) \cdot C_j$$

$$+ \sum_{j \in hp_t(i)} \left(1 + \left\lfloor \frac{min(W_i(t); t+D_i-D_j)+J_j}{T_j} \right\rfloor\right) \cdot C_j$$

$$+ \left\lfloor \frac{t+J_i}{T_i} \right\rfloor \cdot C_i + \max(0\,;\, C_{\overline{max}_{i,t}} - 1).$$

Proof: To compute the latest starting time of packet m, we identify the busy period of level P_i that affect the delay of m, that is this in which m is processed. We define f as the first packet processed in this busy period with a priority greater than or equal to P_i. Due to the non-preemptive effect, the execution of f can be delayed once by a packet in $lp(i) \cup \overline{hp_t}(i)$.

For the sake of simplicity, we number consecutively the packets of the considered busy period of level P_i. Hence, we denote $m' - 1$ (resp. $m' + 1$) the packet preceding (resp. succeeding) m'. Moreover, we consider the release time of packet f as the time origin. The latest starting time of packet m is then equal to the processing time of packets f to $m - 1$, plus $\max(0\,;\, C_{\overline{max}_{i,t}} - 1)$. Then, we get: $W_i(t) = \sum_{g=f}^{m} C_{\tau(g)} - C_i + \max(0\,;\, C_{\overline{max}_{i,t}} - 1)$.

The term $\sum_{g=f}^{m} C_{\tau(g)}$ is bounded by the maximum workload generated by flows belonging to $gp(i)$ in the interval $I_1 = [-J_i, W_i(t)]$, plus the maximum workload generated by flows $\tau_j \in hp_t(i)$ in the interval $I_2 = [-J_j, min(W_i(t); t + D_i - D_j)]$, plus the maximum workload generated by packets of flow τ_i in the interval $I_3 = [-J_i, t]$. By definition, the maximum workload generated by:

- flows $\tau_j \in gp(i)$ in the interval I_1 is equal to: $\sum_{j \in gp(i)} (1 + \lfloor (W_i(t) + J_j)/T_j \rfloor) \cdot C_j$.
- flows $\tau_j \in hp_t(i)$ in the interval I_2 is equal to: $\sum_{j \in hp_t(i)} (1 + \lfloor (min(W_i(t); t + D_i - D_j) + J_j)/T_j \rfloor) \cdot C_j$.
- flow τ_i in the interval I_3 is equal to: $1 + \lfloor (t + J_i)/T_i \rfloor) \cdot C_i$. ■

2.3 Analysis of the latest starting time

We now focus on the following series that we denote $\mathcal{W}_i(t)$, for any time $t \geq -J_i$. To prove the existence of $W_i(t)$, solution of the equation given in Lemma 3, we show that if Condition 1 is met, $\mathcal{W}_i(t)$ is convergent. Then, Lemmas 4 and 5 show that only a limited set of release times of flow τ_i has to be tested to obtain the latest starting time of a flow packet.

$$\begin{cases} W_i^{(0)}(t) = & \sum_{j\in gp(i)\cup hp_t(i)} C_j \\ & + \left\lfloor \frac{t+J_i}{T_i} \right\rfloor \cdot C_i + \max(0\,;\, C_{\overline{max}_{i,t}} - 1) \\ W_i^{(p+1)}(t) = & \sum_{j\in gp(i)} \left(1 + \left\lfloor \frac{W_i^{(p)}(t)+J_j}{T_j} \right\rfloor\right) \cdot C_j \\ & + \sum\limits_{j\in hp_t(i)} \left(1+\left\lfloor \frac{min(W_i^{(p)}(t);t+D_i-D_j)+J_j}{T_j} \right\rfloor\right) \cdot C_j \\ & + \left\lfloor \frac{t+J_i}{T_i} \right\rfloor \cdot C_i + \max(0; C_{\overline{max}_{i,t}} - 1). \end{cases}$$

Condition 1 *If $U_{gp(i)\cup hp_t(i)} < 1$, where $U_{gp(i)\cup hp_t(i)}$ denotes the utilization factor for the flows belonging to $gp(i) \cup hp_t(i)$, then for any time $t \geq -J_i$ the series $\mathcal{W}_i(t)$ is convergent.*

Proof: For any time $t \geq -J_i$, the series $\mathcal{W}_i(t)$ is a non-decreasing series as the floor function is non-decreasing. Moreover, this series is upper bounded by: $X/(1 - U_{gp(i)\cup hp_t(i)})$, where X equals:

$$\sum_{j\in gp(i)\cup hp_t(i)} \left(1+\frac{J_j}{T_j}\right) \cdot C_j + \left\lfloor \frac{t+J_i}{T_i} \right\rfloor \cdot C_i + \max(0\,;\, C_{\overline{max}_{i,t}} - 1).$$

Indeed, by recurrence, we have: $W_i^{(0)}(t) \leq X$, that is less than or equal to: $X/(1 - U_{gp(i)\cup hp_t(i)})$, assuming $U_{gp(i)\cup hp_t(i)} < 1$. We now assume that the recurrence is true at rank p and show that it is true at rank $p+1$. As $min(W_i^{(p)}(t); t + D_i - D_j) \leq W_i^{(p)}(t)$, we have:

$$\begin{aligned} W_i^{(p+1)}(t) \quad \leq \quad & W_i^{(p)}(t) \cdot \sum_{j\in gp(i)\cup hp_t(i)} \frac{C_j}{T_j} \\ & + \sum_{j\in gp(i)\cup hp_t(i)} \left(1 + \frac{J_j}{T_j}\right) \cdot C_j \\ & + \left\lfloor \frac{t+J_i}{T_i} \right\rfloor \cdot C_i + \max(0; C_{\overline{max}_{i,t}} - 1) \\ \leq \quad & W_i^{(p)}(t) \cdot U_{gp(i)\cup hp_t(i)} + X \\ \leq \quad & X \cdot U_{gp(i)\cup hp_t(i)}/(1 - U_{gp(i)\cup hp_t(i)}) + X \\ \leq \quad & X/(1 - U_{gp(i)\cup hp_t(i)}). \end{aligned}$$

The series $\mathcal{W}_i(t)$ is non-decreasing and upper bounded. Hence, this series is convergent. ■

Lemma 4 *In the uniprocessor case, when all flows are scheduled according to* FP/EDF*, the worst case response time of any flow τ_i is obtained for a packet requested at time $t \in \mathcal{S}$, where $\mathcal{S}$ is the ordered set of times $t = k_j \cdot T_j - J_j - D_i + D_j \geq -J_i$, $j \in hp_t(i) \cup \{i\}$ and $k_j \in \mathbb{N}$.*

Proof: We consider the series $\mathcal{W}_i(t)$ and prove the lemma by recurrence. By assumption, no packet of τ_i can be requested before time $-J_i$. Hence, we only consider times $t \geq -J_i$.

By definition, if t_1 and t_2 are two consecutive times of set $\mathcal{S}$, then we get: $\forall\, j \in hp_t(i) \cup \{i\}$, $\forall\, t' \in [t_1, t_2)$, $\lfloor (t' + D_i - D_j + J_j)/T_j \rfloor = \lfloor (t_1 + D_i - D_j + J_j)/T_j \rfloor$. Therefore, the lemma is met at rank 0. Indeed, $W_i^{(0)}(t') = W_i^{(0)}(t_1)$. Assuming that the recurrence is true at rank p, we show that it is true at rank $p+1$. $W_i^{(p+1)}(t')$ is equal to:

$$
\begin{aligned}
& \textstyle\sum_{j \in gp(i)} \left(1 + \left\lfloor \frac{W_i^{(p)}(t') + J_j}{T_j} \right\rfloor\right) \cdot C_j \\
& + \textstyle\sum_{j \in hp_t(i)} \left(1 + \left\lfloor \frac{min(W_i^{(p)}(t'); t' + D_i - D_j) + J_j}{T_j} \right\rfloor\right) \cdot C_j \\
& + \left\lfloor \frac{t' + J_i}{T_i} \right\rfloor \cdot C_i + \max(0\,;\, C_{\overline{max}_{i,t}} - 1) \\
= \; & \textstyle\sum_{j \in gp(i)} \left(1 + \left\lfloor \frac{W_i^{(p)}(t_1) + J_j}{T_j} \right\rfloor\right) \cdot C_j \\
& + \textstyle\sum_{j \in hp_t(i)} \left(1 + \left\lfloor \frac{min(W_i^{(p)}(t_1); t_1 + D_i - D_j) + J_j}{T_j} \right\rfloor\right) \cdot C_j \\
& + \left\lfloor \frac{t_1 + J_i}{T_i} \right\rfloor \cdot C_i + \max(0\,;\, C_{\overline{max}_{i,t}} - 1) \\
= \; & W_i^{(p+1)}(t_1). \qquad \blacksquare
\end{aligned}
$$

Lemma 5 *In the uniprocessor case, when all flows are scheduled according to* FP/EDF*, if a packet of τ_i requested at time $t \in \mathcal{S}$ meets: $W_i(t) + C_i \leq t + T_i$, then it is useless to compute the response time of a packet requested at time $t + \alpha \cdot T_i$, $\alpha \in \mathbb{N}^\star$.*

Proof: Let us consider a fixed $j \in hp_t(i) \cup \{i\}$. Let m' be the packet of flow τ_i requested at time $t \in \mathcal{S}$. We compute the response time of m' and distinguish two cases:

- If $W_i(t) + C_i > t + T_i$, the packet of τ_i requested at time $t + T_i$ is delayed by the processing of previous packets of τ_i. Hence, the response time of this packet must be computed.
- If $W_i(t) + C_i \leq t + T_i$, the packet of τ_i requested at time $t + T_i$ is processed without being delayed by previous packets of τ_i. According to Lemma 2, this packet is not

processed in the worst case conditions. Hence, the worst case response time of τ_i will not be reached by this packet. ■

2.4 Worst case response time

With the previous lemmas, we can now establish the following property, that gives the worst case response time of any flow τ_i.

Property 1 *In the uniprocessor case, when all flows are scheduled according to* FP/EDF *and condition 1 is met, the worst case response time of any flow* τ_i *meets:* $R_i = \max_{t \in \mathcal{S}'} \{W_i(t) - t\} + C_i$*, where:*

$$W_i(t) = \sum_{j \in gp(i)} \left(1 + \left\lfloor \frac{W_i(t)+J_j}{T_j} \right\rfloor\right) \cdot C_j$$
$$+ \sum_{j \in hp_t(i)} \left(1 + \left\lfloor \frac{min(W_i(t); t+D_i-D_j)+J_j}{T_j} \right\rfloor\right) \cdot C_j$$
$$+ \left\lfloor \frac{t+J_i}{T_i} \right\rfloor \cdot C_i + \max(0\,; C_{\overline{max}_{i,t}} - 1)$$

and $\mathcal{S}'$ *denotes the ordered set of times* t *equal to:* $t = k_j \cdot T_j - D_i + D_j - J_j \geq -J_i$*, excluding the times* $t' = k_j \cdot T_j - D_i + D_j - J_j + \alpha \cdot T_i$ *such that* $W_i(t') + C_i \leq t' + T_i$*, with* $j \in hp_t(i) \cup \{i\}$*,* $k_j \in \mathbb{N}$ *and* $\alpha \in \mathbb{N}^\star$*.*

Proof: From Lemmas 3, 4 and 5. ■

3 Comparative evaluation for a uniprocessor

In this section, we first recall the computation of the response time of any flow when scheduled with classical Fixed Priority. We then show that any set of sporadic flows feasible with Fixed Priority scheduling is feasible with FP/EDF scheduling (Property 3), but the converse is false (Property 4), as illustrated by an example.

3.1 Classical results for FP scheduling

Classical results have been established assuming that flows sharing the same priority are scheduled arbitrarily. We recall them here, in order to make a comparison with our results, established when flows sharing the same priority are scheduled EDF. A packet m of flow τ_i, generated at time t, can be delayed by packets of flows in $gp(i) \cup sp(i)$ arrived at a time less than or equal to $W_i(t)$ and a packet of a flow in $lp(i)$.

In order to easily compare the response times of a flow τ_i obtained with Fixed Priority and with FP/EDF in Subsection 3.3, we present the classical results obtained with Fixed Priority in a way similar to Property 1.

Property 2 *In the uniprocessor case, when flows are scheduled according to the Fixed Priority algorithm, the worst case response time R_i of any flow τ_i meets: $R_i = \max_{t \in \mathcal{S}} \{W_i(t) - t\} + C_i$, where:*

$$W_i(t) = \sum_{j \in gp(i) \cup sp(i)} \left(1 + \left\lfloor \frac{W_i(t)+J_j}{T_j} \right\rfloor\right) \cdot C_j$$
$$+ \left\lfloor \frac{t+J_i}{T_i} \right\rfloor \cdot C_i + \max\left(0\,;\, C_{\overline{max},i} - 1\right),$$

and $\mathcal{S}$ denotes the ordered set of times $t = k \cdot T_i - J_i$, $k = 0..K$, with K the smallest integer value such that $W_i(K \cdot T_i - J_i) + C_i + J_i \leq (K+1) \cdot T_i$.

Proof: See [1, 4]. ■

3.2 Example

In this example, we consider five flows, without release jitter, whose characteristics are given in Table 1. The load is equal to 100%. There is only one flow in the priorities 3 and 2, whereas three flows share the lowest priority 1 with respective deadlines 26, 28 and 30. We then obtain the following results (see Table 1) for the worst case response time of any flow.

Table 1: Improvement on the worst case response time

					FP/EDF		*FP*	
Flow	P_i	C_i	T_i	D_i	$W_i(t)$	R_i	$W_i(t)$	R_i
τ_1	1	4	20	26	20	24	32	36
τ_2	1	4	20	28	22	26	32	36
τ_3	1	4	20	30	24	28	32	36
τ_4	2	4	20	15	11	15	11	15
τ_5	3	8	40	11	3	11	3	11

As expected, results show that for priorities 2 and 3, there is no difference between FP/EDF and Fixed Priority. Indeed, there is only one flow for each of these two priorities. Concerning the lowest priority, FP/EDF gives a worst case response time of 24, whereas Fixed Priority gives 36. FP/EDF gets an improvement higher than 33%.

As with FP/EDF, all flows meet their deadline, this flow set is feasible with FP/EDF. However, it is not feasible with classical Fixed Priority as the flows τ_1, τ_2 and τ_3 do not meet their deadlines respectively 26, 28 and 30. This example shows the interest of FP/EDF.

3.3 Comparison between FP/EDF and FP

We establish properties 3 and 4 that highlight the benefit brought by an EDF scheduling of flows sharing the same priority.

Property 3 *Any sporadic flow set feasible with Fixed Priority scheduling is feasible with* FP/EDF.

Proof: As FP accounts for any arbitrary scheduling of flows sharing the same fixed priority, it accounts for EDF scheduling. Hence property 3. ■

Property 4 *A sporadic flow set feasible with* FP/EDF *can be not feasible with Fixed Priority scheduling.*

Proof: We now show that there exist packets that can delay m with Fixed Priority scheduling but not with FP/EDF. These packets are those in $\overline{hp}_t(i)$, except (if any) this having the maximum duration higher than any packet in $lp(i)$. The example given in Section 3.2 shows that the converse of property 3 is false. ■

Properties 3 and 4 highlight the benefits of scheduling EDF flows belonging to the same class (i.e. sharing the same fixed priority).

3.4 Comparison between FP/FIFO and FP

As FIFO scheduling can be seen as a particular case of EDF scheduling where all flows have the same deadline, we can deduce the worst case response time obtained with FP/FIFO from this obtained with FP/EDF (see Property 1). We then establish Properties 6 and 7 that highlight the benefit brought by a FIFO scheduling of flows sharing the same priority.

Property 5 *In the uniprocessor case, when all flows are scheduled according to* FP/FIFO *and condition 1 is met, the worst case response time of any flow* τ_i *meets:* $R_i = \max_{t \in \mathcal{S}'} \{W_i(t) - t\} + C_i$*, where:*

$$W_i(t) = \sum_{j \in gp(i)} \left(1 + \left\lfloor \frac{W_i(t)+J_j}{T_j} \right\rfloor\right) \cdot C_j + \sum_{j \in sp(i)} \left(1 + \left\lfloor \frac{t+J_j}{T_j} \right\rfloor\right) \cdot C_j + \left\lfloor \frac{t+J_i}{T_i} \right\rfloor \cdot C_i$$
$$+ \max(0\,; C_{\overline{max},i} - 1) \text{ with } C_{\overline{max},i} = max_{j \in lp(i)}\{C_j\} \text{ if } lp(i) \neq \emptyset \text{ and } 0 \text{ otherwise,}$$

and $\mathcal{S}'$ *denotes the ordered set of times* t *equal to:* $k_j \cdot T_j - J_j \geq -J_i$*, excluding the times* $t' = k_j \cdot T_j - J_j + \alpha \cdot T_i$ *such that* $W_i(t') + C_i \leq t' + T_i$*,* $j \in sp(i) \cup \{i\}$*,* $k_j \in \mathbb{N}$ *and* $\alpha \in \mathbb{N}^\star$*.*

Proof: With FIFO, $D_i = D_j$ for any flows τ_i and τ_j. Consequently: $\forall\, t,\ hp_t(i) = sp(i)$. Moreover, by definition, we have: $\forall\, t,\ W_i(t) \geq t$. Then, we get: $min(W_i(t); t + D_i - D_j) = t$. Hence Property 5. ■

Notice that if all flows having the same fixed priority have the same processing time and the same release jitter, they have the same worst case response time.

Property 6 *Any sporadic flow set feasible with Fixed Priority scheduling is feasible with* FP/FIFO.

Proof: As FP accounts for any arbitrary scheduling of flows sharing the same fixed priority, it accounts for FIFO scheduling. Hence property 6. ■

Property 7 *A sporadic flow set feasible with* FP/FIFO *can be not feasible with Fixed Priority scheduling.*

Proof: This is proved by the sporadic flow set given in Table 2. The worst case response times obtained with Fixed Priority and FP/FIFO are given in Table 2.

Table 2: Worst case response times with FP and FP/FIFO

					FP/FIFO		*FP*	
Flow	P_i	C_i	T_i	D_i	$W_i(t)$	R_i	$W_i(t)$	R_i
τ_1	1	4	20	28	24	28	32	36
τ_2	1	4	20	28	24	28	32	36
τ_3	1	4	20	28	24	28	32	36
τ_4	2	4	20	15	11	15	11	15
τ_5	3	8	40	11	3	11	3	11

This flow set meets its deadlines with FP/FIFO but not with Fixed Priority. ■

3.5 FP/EDF versus FP/FIFO

We consider a sporadic flow set such that all flows sharing the same priority have the same processing time. This assumption is realistic as the processing time of a packet on a node mainly depends on its Fixed Priority. For instance, in a DiffServ architecture, the processing time of a packet depends on its class identified by its DiffServ code.

Property 8 *Any sporadic flow set such that all flows sharing the same priority have the same processing time, feasible with* FP/FIFO *is feasible with* FP/EDF.

Proof: We prove that if such a flow set is not feasible with FP/EDF, then this set is not feasible with FP/FIFO. We assume that all flows are scheduled FP/EDF and there exists a packet m of a flow τ_i that does not meet its deadline. We will show that there exists a packet with the same priority that would not meet its deadline if flows were scheduled FP/FIFO.

We focus on packet m of flow τ_i requested at time t that does not meet its deadline. Let $W_i^{FP/EDF}(t)$ be the latest starting time of m. As m does not meet its deadline, we have $W_i^{FP/EDF}(t) + C_i > t + D_i$, with $W_i^{FP/EDF}(t) + C_i$ equals to:

$$\sum_{j \in gp(i)} \left(1 + \left\lfloor \frac{W_i^{FP/EDF}(t)+J_j}{T_j} \right\rfloor \right) \cdot C_j$$

$$+ \sum_{j \in hp_t(i)} \left(1 + \left\lfloor \frac{min(W_i^{FP/EDF}(t); t+D_i-D_j)+J_j}{T_j} \right\rfloor \right) \cdot C_j$$

$$+ \left(1 + \left\lfloor \frac{t+J_i}{T_i} \right\rfloor \right) \cdot C_i + \max(0\,;\, C_{\overline{max}_{i,t}} - 1).$$

As $W_i^{FP/EDF}(t) + C_i > t + D_i$, we have:

$\min_{j \in hp_t(i)}(W_i^{FP/EDF}(t) + C_i; t + D_i - D_j) = t + D_i - D_j$.

Let τ_k be the flow such that $D_k = \min\limits_{\substack{j \in sp(i) \cup \{i\} \\ t+D_i-D_j \geq -J_j}} \{D_j\}$, that is the flow having the smallest deadline among the flows τ_j having a priority equal to P_i such that: $t + D_i - D_j \geq -J_j$.

Moreover, we define $\alpha = t + D_i - D_k$ and $\forall j \in sp(i) \cup \{i\}$, $\Delta_j = D_j - D_k$. Then, $W_i^{FP/EDF}(t) + C_i = W_i^{FP/EDF}(\alpha - \Delta_i) + C_i$, that is:

$$\sum_{j \in gp(i)} \left(1 + \left\lfloor \frac{W_i^{FP/EDF}(\alpha-\Delta_i)+J_j}{T_j} \right\rfloor \right) \cdot C_j$$

$$+ \sum_{j \in hp_{\alpha-\Delta_i}(i)} \left(1 + \left\lfloor \frac{\alpha-\Delta_j+J_j}{T_j} \right\rfloor \right) \cdot C_j$$

$$+ \left(1 + \left\lfloor \frac{\alpha-\Delta_i+J_i}{T_i} \right\rfloor \right) \cdot C_i + \max(0\,;\, C_{\overline{max}_{i,\alpha-\Delta_i}} - 1).$$

We now show that if flows were scheduled FP/FIFO, a packet m' of flow τ_k requested at time α would not meet its deadline. Let $W_i^{FP/FIFO}(\alpha)$ be the latest starting time of m' when flows are scheduled FP/FIFO. We proceed in four steps.

• **First step**: We first establish the following result:

$$\begin{aligned} & \sum_{j\in hp_{\alpha-\Delta_i}(i)} \left(1+\left\lfloor\frac{\alpha-\Delta_j+J_j}{T_j}\right\rfloor\right)\cdot C_j \\ & +\left(1+\left\lfloor\frac{\alpha-\Delta_i+J_i}{T_i}\right\rfloor\right)\cdot C_i \\ & +\max(0\,;\,C_{\overline{max}_{i,\alpha-\Delta_i}}-1) \\ \leq\ & \sum_{j\in sp(i)\cup\{i\}} \left(1+\left\lfloor\frac{\alpha+J_j}{T_j}\right\rfloor\right)\cdot C_j \\ & +\max(0\,;\,C_{\overline{max},i}-1). \end{aligned}$$

We show that any flow counted in the left part of the inequation is counted in the right part. We first consider the task of maximum execution duration in $lp(i)\cup\overline{hp}_{\alpha-\Delta_i}(i)$, if any. It is counted in the left part. We distinguish two cases:

- it belongs to $lp(i)$: it is then counted in the right part;
- it belongs to $\overline{hp}_{\alpha-\Delta_i}(i)$: as it also belongs to $sp(i)$ it is counted in the first term of the right part.

Moreover, as $\left\lfloor\frac{\alpha-\Delta_j+J_j}{T_j}\right\rfloor\leq\left\lfloor\frac{\alpha+J_j}{T_j}\right\rfloor$ and $hp_{\alpha-\Delta_i(i)}\subseteq sp(i)$, we obtain the inequation considered.

• **Second step**: We now prove by recurrence that: $W_i^{FP/EDF}(\alpha-\Delta_i)\leq W_i^{FP/FIFO}(\alpha)$.

For this, we consider the associated series and proceed by recurrence.

At rank 0, we have $W_i^{(0)FP/EDF}(\alpha-\Delta_i)$ equals to:

$$\begin{aligned} & \sum_{j\in gp(i)\cup hp_{\alpha-\Delta_i}(i)} C_j+\left\lfloor\frac{\alpha-\Delta_i+J_i}{T_i}\right\rfloor\cdot C_i \\ & +\max(0\,;\,C_{\overline{max}_{i,\alpha-\Delta_i}}-1) \\ \leq\ & \sum_{j\in gp(i)\cup sp(i)} C_j+\left\lfloor\frac{\alpha+J_i}{T_i}\right\rfloor\cdot C_i \\ & +\max(0\,;\,C_{\overline{max},i}-1) \\ \leq\ & W_i^{(0)FP/FIFO}(\alpha). \end{aligned}$$

We assume that $W_i^{(p)FP/EDF}(\alpha-\Delta_i)$ is less than or equal to $W_i^{(p)FP/FIFO}(\alpha)$ and we consider $W_i^{(p+1)FP/EDF}(\alpha-\Delta_i)$, that is equal to:

$$\begin{aligned}
& \textstyle\sum_{j\in gp(i)}\left(1+\left\lfloor\frac{W_i^{(p)FP/EDF}(\alpha-\Delta_i)+J_j}{T_j}\right\rfloor\right)\cdot C_j \\
& +\textstyle\sum_{j\in hp_{\alpha-\Delta_i}(i)}\left(1+\left\lfloor\frac{\alpha-\Delta_j+J_j}{T_j}\right\rfloor\right)\cdot C_j \\
& +\left\lfloor\frac{\alpha-\Delta_i+J_i}{T_i}\right\rfloor\cdot C_i+\max(0\,;\,C_{\overline{max}_{i,\alpha-\Delta_i}}-1) \\
\leq\; & \textstyle\sum_{j\in gp(i)}\left(1+\left\lfloor\frac{W_i^{(p)FP/FIFO}(\alpha)+J_j}{T_j}\right\rfloor\right)\cdot C_j \\
& +\textstyle\sum_{j\in sp(i)\cup\{i\}}\left(1+\left\lfloor\frac{\alpha+J_j}{T_j}\right\rfloor\right)\cdot C_j-C_i \\
& +\max(0\,;\,C_{\overline{max},i}-1) \\
\leq\; & W_i^{(p+1)FP/FIFO}(\alpha).
\end{aligned}$$

As both series are convergent, their limits meet $W_i^{FP/EDF}(\alpha-\Delta_i)\leq W_i^{FP/FIFO}(\alpha)$. Hence, $W_i^{FP/FIFO}(\alpha)+C_i\geq t+D_i=\alpha+D_k$.

• **Third step**: Moreover, we can show than when flows having the same fixed priority have the same processing time, flow τ_i and flow τ_k experience the same latest starting time with FP/FIFO, when requested at time α.
Notice that $\alpha\geq -J_i$, since $D_i\geq D_k$ and $\alpha\geq -J_k$ by construction. We want to show $W_i^{FP/FIFO}(\alpha)=W_k^{FP/FIFO}(\alpha)$.
We can write $W_i^{FP/FIFO}(\alpha)+C_i$ as follows:

$$\begin{aligned}
W_i^{FP/FIFO}(\alpha)+C_i \;=\;& \textstyle\sum_{j\in gp(i)}\left(1+\left\lfloor\frac{W_i^{FP/FIFO}(\alpha)+J_j}{T_j}\right\rfloor\right)\cdot C_j \\
& +\textstyle\sum_{j\in sp(i)\cup\{i\}}\left(1+\left\lfloor\frac{\alpha+J_j}{T_j}\right\rfloor\right)\cdot C_j \\
& +\max(0\,;\,C_{\overline{max},i}-1).
\end{aligned}$$

We notice the following equalities: $gp(i)=gp(k)$, $lp(i)=lp(k)$ and $sp(i)\cup\{i\}=sp(k)\cup\{k\}$. Hence, we get $\max(0\,;\,C_{\overline{max},i}-1)=\max(0\,;\,C_{\overline{max},k}-1)$.
As $C_i=C_k$, we prove by recurrence $W_i^{FP/FIFO}(\alpha)=W_k^{FP/FIFO}(\alpha)$. At rank 0, we have:

$$\begin{aligned} W_i^{FP/FIFO(0)}(\alpha) + C_i &= \sum_{j\in gp(i)} C_i + \sum_{j\in sp(i)\cup\{i\}} \left(1 + \left\lfloor \frac{\alpha+J_j}{T_j} \right\rfloor\right) \cdot C_j + \max(0\,;\, C_{\overline{max},i} - 1) \\ &= W_k^{FP/FIFO(0)}(\alpha) + C_k. \end{aligned}$$

Assuming that $W_i^{FP/FIFO(p)}(\alpha) + C_i = W_k^{FP/FIFO(p)}(\alpha) + C_k$, we prove it at rank $p+1$.

$$\begin{aligned} W_i^{FP/FIFO(p+1)}(\alpha) + C_i &= \textstyle\sum_{j\in gp(i)} \left(1 + \left\lfloor \frac{W_i^{FP/FIFO(p)}(\alpha)+J_j}{T_j} \right\rfloor\right) \cdot C_j \\ &\quad + \textstyle\sum_{j\in sp(i)\cup\{i\}} \left(1 + \left\lfloor \frac{\alpha+J_j}{T_j} \right\rfloor\right) \cdot C_j + \max(0\,;\, C_{\overline{max},i} - 1) \\ &= \textstyle\sum_{j\in gp(k)} \left(1 + \left\lfloor \frac{W_k^{FP/FIFO(p)}(\alpha)+C_k-C_i+J_j}{T_j} \right\rfloor\right) \cdot C_j \\ &\quad + \textstyle\sum_{j\in sp(k)\cup\{k\}} \left(1 + \left\lfloor \frac{\alpha+J_j}{T_j} \right\rfloor\right) \cdot C_j + \max(0\,;\, C_{\overline{max},k} - 1) \\ &= W_k^{(p+1)}(\alpha) + C_k. \end{aligned}$$

Hence $W_i^{FP/FIFO}(\alpha) = W_k^{FP/FIFO}(\alpha)$.

• **Fourth step**: From the previous steps, we have:

$$\begin{aligned} W_k^{FP/FIFO}(\alpha) + C_k &= W_i^{FP/FIFO}(\alpha) + C_i \\ &\geq W_i^{FP/EDF}(t) + C_i \\ &> t + D_i = \alpha + D_k. \end{aligned}$$

Therefore, packet m' of flow τ_k does not meet its deadline when flows are scheduled FP/FIFO. Hence, when flows sharing the same fixed priority have the same processing time, if a packet m of flow τ_i does not meet its deadline when flows are scheduled FP/EDF, then there exists a packet m' of flow τ_k, $k \in sp(i) \cup \{i\}$, that does not meet its deadline when flows are scheduled FP/FIFO. ■

Property 9 *A sporadic flow set such that all flows sharing the same priority have the same processing time, feasible with* FP/EDF *can be not feasible with* FP/FIFO.

Proof: This is proved by the sporadic flow set given in Table 1. This flow set meets its deadlines with FP/EDF (see Table 1) but not with FP/FIFO (see the response time of flow τ_1 in Table 2). ■

As a consequence of properties 8 and 9, FP/EDF dominates FP/FIFO when flows sharing the same priority have the same processing time.

4 Response time computation in the distributed case

To determine the maximum end-to-end response time, several approaches can be used: a stochastic or a deterministic one. A *stochastic approach* consists in determining the mean behavior of the considered network, leading to mean, statistical or probabilistic end-to-end response times [8, 9]. A *deterministic approach* is based on a worst case analysis of the network behavior, leading to worst case end-to-end response times [10, 11].

In this paper, we are interested in the deterministic approach as we want to provide a deterministic guarantee of worst case end-to-end response times for any flow in the network. In this context, two different approaches can be used to determine the worst case end-to-end delay: the holistic approach and the trajectory approach.

• The *holistic approach* [12] considers the worst case scenario on each node visited by a flow, accounting for the maximum possible jitter introduced by the previous visited nodes. If no jitter control is done, the maximum jitter will increase throughout the visited nodes. In this case, the minimum and maximum response times on a node h induce a maximum jitter on the next visited node $h+1$ that leads to a worst case response time and then a maximum jitter on the following node and so on. Otherwise, the jitter can be either cancelled or constrained.

- ◦ the *Jitter Cancellation technique* consists in cancelling, on each node, the jitter of a flow before it is considered by the node scheduler [11]: a flow packet is held until its latest possible reception time. Hence a flow packet arrives at node $h+1$ with a jitter depending only on the jitter introduced by the previous node h and the link between them. As soon as this jitter is cancelled, this packet is seen by the scheduler of node $h+1$. The worst case end-to-end response time is obtained by adding the worst case response time, without jitter (as cancelled) on every node;
- ◦ the *Constrained Jitter technique* consists in checking that the jitter of a flow remains bounded by a maximum acceptable value before the flow is considered by the node scheduler. If not, the jitter is reduced to the maximum acceptable value by means of traffic shaping.

As a conclusion, the holistic approach can be pessimistic as it considers worst case scenarios on every node possibly leading to impossible scenarios.

• The *trajectory approach* [13] consists in examining the scheduling produced by all the visited nodes of a flow. In this approach, only possible scenarios are examined. For instance, the fluid model (see [14] for GPS) is relevant to the trajectory approach. This approach produces the best results as no impossible scenario is considered but is somewhat more complex to use. This approach can also be used in conjunction with a jitter control (see [15] for EDF, and [14] for GPS). In this paper, we adopt the trajectory approach without jitter control in a distributed system to determine the maximum end-to-end response time of a flow.

We can also distinguish two main traffic models: the sporadic model and the token bucket model. The sporadic model has been used in the holistic approach and in the trajectory approach, while the token bucket model has been used only in the trajectory approach.

• The *sporadic model* is classically defined by three parameters: the maximum processing time, the minimum interarrival time and the maximum release jitter, (see the traffic model in section 5). This model is natural and well adapted for real-time applications.

• The *token bucket* [10, 14, 15] is defined by two parameters: σ, the bucket size and ρ, the token throughput. The token bucket can model a flow or a flow aggregate. In the first case, it requires to maintain per flow information on every visited node. This solution is not scalable. In the second case, the choice of good values for the token bucket parameters is complex when flows have different characteristics. A bad choice can lead to bad response times, as the end-to-end response times strongly depend on the choice of the token bucket parameters [15, 16]. Furthermore, the token bucket parameters can be optimized for a given configuration, only valid at a given time. If the configuration evolves, the parameters of the token bucket should be recomputed on every node to remain optimal. This is not generally done.

In this paper, we adopt the trajectory approach with the sporadic traffic model and we establish new results that we compare with those provided by the classical holistic approach.

5 FP/EDF scheduling in a distributed context: Line case

We investigate the problem of providing a deterministic end-to-end response time guarantee to any flow in a distributed system, when these sporadic flows are scheduled FP/EDF. The end-to-end response time of a flow is defined between its ingress node and its egress node. We want to provide an upper bound on the end-to-end response time of any flow. As we make no particular assumption concerning the arrival times of packets in the distributed system, the feasibility of a set of flows is equivalent to meet the requirement, whatever the arrival times of the packets in the distributed system. Moreover, we assume the following models.

Network model: We consider a distributed system where links interconnecting nodes are supposed to be FIFO and the network delay between two nodes has known lower and upper bounds: L_{min} and L_{max}. Moreover, we consider neither network failures nor packet losses.

Traffic model: We consider a set $\tau = \{\tau_1, ..., \tau_n\}$ of n sporadic flows. Each flow τ_i follows a sequence of nodes whose first node is the ingress node of the flow. In the following, we call *line* this sequence. Moreover, a sporadic flow τ_i following a line $\mathcal{L}$ consisted of q nodes, numbered from 1 to q, is defined by:

- T_i, the minimum interarrival time (called period) between two successive packets of τ_i;
- C_i^h, the maximum processing time on node h of a packet of τ_i;
- J_i^1, the maximum jitter of packets of τ_i arriving in the distributed system;
- D_i, the end-to-end deadline of any packet of τ_i.

Scheduling model: We consider that all nodes in the network schedule packets according to FP/EDF. Moreover, we assume that packet scheduling is non-preemptive. Therefore, the node scheduler waits for the completion of the current packet transmission (if any) before selecting the next packet.

For any flow τ_i, when a packet of τ_i, requested at time t, arrives on node 1, the first node visited, it is marked with its static priority P_i and its absolute deadline, equal to $t + D_i^1$, where D_i^1 is the relative deadline of flow τ_i on node 1. This solution presents the additional advantage of not requiring clock synchronization in the core nodes. Indeed, only the ingress nodes require clock synchronization to assign to each incoming packet its absolute deadline. The computation of D_i^1 is made from the end-to-end deadline D_i. A flow τ_i having an end-to-end deadline D_i but visiting few nodes must have an intermediate deadline D_i^1 higher than a flow τ_j with the same end-to-end deadline but visiting more nodes. A simple way consists in selecting $D_i^1 = \lfloor D_i/q \rfloor$. Other solutions exist, for instance solutions accounting for the workload of the visited nodes. The selection of the best value is out of the scope of this paper.

Each node visited by this packet schedules it according to FP/EDF accounting for its fixed priority P_i and its absolute deadline $t + D_i^1$, computed on node 1. This scheduling rule ensures that for any packet m of τ_i, requested at time t on node 1, all visited nodes take their scheduling decision on exactly the same values of both the fixed priority P_i and the absolute deadline $t + D_i^1$. Notice that EDF is applied on $t + D_i^1$ on each visited node, but the end-to-end deadline that must be met by flow τ_i is $t + D_i$.

Property 10 *FP/EDF ensures that for any packets m and m', if m has a generalized priority higher than m' on a node, then this is true on any visited node.*

Notice that if each node applies FP/EDF based on a local intermediate deadline, then Property 10 is not met. This is proved by the following example. We consider two sporadic flows (τ_1 and τ_2), without jitter when entering the distributed system and visiting two nodes. Each node applies a local EDF. Table 3 provides the characteristics of the flows considered, with D_i^1 (respectively D_i^2) the intermediate deadline of τ_i, $i \in [1, 2]$, on the first (respectively last) node. Finally, we assume that $L_{min} = L_{max} = 1$. Let m and m' be respectively the first packet of flow τ_1 and τ_2, entered the system at time 0. Packet m' has an intermediate absolute deadline smaller than packet m in the first node $(3 < 6)$, but not in the last node $(3 + 7 > 5 + 4)$. Hence Property 10 is not met.

Table 3: Characteristics of flows τ_1, τ_2 and τ_3

Flow	P_i	C_i^1	C_i^2	T_i	D_i	D_i^1	D_i^2
τ_1	1	2	2	10	10	6	4
τ_2	1	2	2	10	10	3	7

Moreover, notice that Property 10 does not imply that the scheduling order is the same on all nodes. Indeed, let us consider the case illustrated by Figure 1, where the generalized priority of a packet m' is higher than this of m. Suppose that node 1 is idle when m arrives. Packet m is then immediately processed. Packet m', arrived after, has to wait. Suppose that node 2 is busy when m and m' arrive. Packet m' is then processed before packet m according to their priorities.

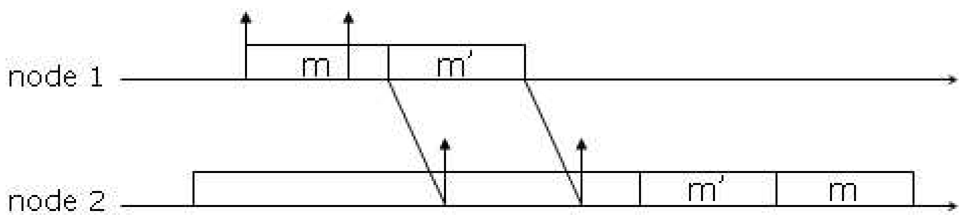

Figure 1. The scheduling order depends on the node

5.1 Methodology

We consider that all flows follow the same line $\mathcal{L}$ in the distributed system, that is the same sequence of nodes consisting of q nodes numbered from 1 to q. We want to determine the end-to-end response time of any packet m, requested at time t and belonging to any flow τ_i. With non-preemptive scheduling, if a packet arrives on any visited node h after m starts its processing, then it cannot delay m. Hence we compute the latest starting time of m on node q, the last node visited (subsection 5.2). Then, the mathematical expression of this latest starting time, that is an iterative expression, is analyzed (subsection 5.3). Finally, we deduce from this time a bound on the end-to-end response time (subsection 5.4). Moreover, we show how to get a more accurate value of the delay due to packets with a priority less than m (subsection 5.5). We first define our notations.

Notations and preliminary results

For any flow τ_i, we denote:

- $gp(i) = \{j \in [1, n]$, such that $P_j > P_i\}$;
- $sp(i) = \{j \in [1, n], j \neq i$, such that $P_j = P_i\}$;
- $lp(i) = \{j \in [1, n]$, such that $P_j < P_i\}$.

For any packet m of flow τ_i, requested at time t on node 1, we denote:

- $hp_t(i) = \{j \in sp(i)$, such that $D_j^1 - J_j^1 \leq t + D_i^1\}$;
- $\overline{hp}_t(i) = \{j \in sp(i)$, such that $D_j^1 - J_j^1 > t + D_i^1\}$.

In addition to these notations, we use the following ones:

$\mathcal{L}$	line consisting of q nodes numbered from 1 to q and followed by all the flows;
$slow$	the slowest node of line $\mathcal{L}$, that is: $\forall$ flow τ_j, $\forall$ node $h \in \mathcal{L}$, $C_j^h \leq C_j^{slow}$;
$W_i^q(t)$	the latest starting time on node q of the packet of τ_i requested at time t on node 1;
$R_i^{\mathcal{L}}$	the worst case end-to-end response time of flow τ_i in the distributed system;
$M_i^{1,q}$	the minimum time taken by a packet of flow τ_i to go from node 1 to node q;
$L_{m'}^{h,h+1}$	the network delay experienced by the packet m' between nodes h and $h+1$;
$H_i^{1,h}(t)$	the maximum delay incurred by the packet of flow τ_i, requested at time t on node 1, directly due to flows $\in lp(i) \cup \overline{hp}_t(i)$, while visiting nodes 1 to h.

Moreover, on any node h, the processing time of any packet m' is less than or equal to:

- $C_{max_{i,t}}^h = \max_{j \in gp(i) \cup hp_t(i) \cup \{i\}} \{C_j^h\}$ if $m' \in gp(i) \cup hp_t(i) \cup \{i\}$;
- $C_{\overline{max}_{i,t}}^h = \max_{j \in lp(i) \cup \overline{hp}_t(i)} \{C_j^h\}$ if $m' \in lp(i) \cup \overline{hp}_t(i)$. If $lp(i) \cup \overline{hp}_t(i) = \emptyset$, then $C_{\overline{max}_{i,t}}^h = 0$.

Finally, we denote $P_{i,t+D_i^1}$ the generalized priority of the packet of flow τ_i requested at time t on node 1, with an absolute deadline equal to $t + D_i^1$ and a fixed priority P_i. In a sake of concision, we say that:

- packet m of flow τ_i, requested at time t on node 1, has priority $P_{i,t+D_i^1}$;
- packet m' of flow τ_j,with $j \neq i$, requested at time t' on node 1, has a priority:
    - higher than m if and only if either $P_j > P_i$ or $P_j = P_i$ and $t' + D_j^1 \leq t + D_i^1$;
    - smaller than m if and only if either $P_j < P_i$ or $P_j = P_i$ and $t' + D_j^1 > t + D_i^1$.

We now give the definition of a level $P_{i,t+D_i^1}$ busy period accounting for the fixed priority P_i and deadline $t + D_i^1$.

Definition 6 *An idle time t' of level $P_{i,t+D_i^1}$ is a time such that all packets with a priority greater than or equal to $P_{i,t+D_i^1}$ and arrived before t' have been processed at time t'.*

Definition 7 *A level $P_{i,t+D_i^1}$ busy period is defined by an interval $[t', t'')$ such that t' and t'' are both idle times of level $P_{i,t+D_i^1}$ and there is no idle time of level $P_{i,t+D_i^1} \in (t', t'')$.*

On any node h, a level $P_{i,t+D_i^1}$ busy period contains only packets belonging to $gp(i) \cup hp_t(i) \cup \{i\}$, except the first one directly due to the non-preemption.

With non-preemptive scheduling on any node h, any packet m of flow τ_i, requested at time t on node 1, can be delayed by a packet m' with a smaller priority if m' has started its processing before the arrival on node h of m. The following lemma gives an upper bound on the maximum delay incurred by m and directly due to packets belonging to $lp(i) \cup \overline{hp}_t(i)$.

Lemma 6 *When all flows are scheduled FP/EDF and follow the same line $\mathcal{L}$ consisting of q nodes numbered from 1 to q, then the maximum delay incurred by the packet m of any flow τ_i, requested at time t on node 1, directly due to packets belonging to $lp(i) \cup \overline{hp}_t(i)$ meets: $H_i^{1,q}(t) \leq \sum_{h=1}^{q} \max(0\,; C^h_{\overline{max}_{i,t}} - 1)$, where $C^h_{\overline{max}_{i,t}}$ denotes the maximum processing time on node h of a packet with a priority smaller than $P_{i,t+D_i^1}$.*

Proof: By definition, no packet of flows belonging to $lp(i) \cup \overline{hp}_t(i)$ can be processed in a busy period of level $P_{i,t+D_i^1}$, except the first packet of this busy period (see Section 5.2). Hence, the maximum delay incurred by any packet m of flow τ_i directly due to packets belonging to $lp(i) \cup \overline{hp}_t(i)$ cannot be greater than $\max(0\,; C^h_{\overline{max}_{i,t}} - 1)$ on each node h visited. ■

5.2 Computation of the latest starting time

Lemma 7 *When all flows follow the same line $\mathcal{L}$ consisting of q nodes numbered from 1 to q and are scheduled according to FP/EDF, then for any packet belonging to any flow τ_i, requested at time t on node 1, its latest starting time in node q is given by:*

$$W_i^q(t) = \sum_{j\in gp(i)} \left(1 + \left\lfloor \frac{\max\left(0\,;\, W_i^q(t) - M_j^{1,q}\right) + J_j^1}{T_j} \right\rfloor\right) \cdot C_j^{slow} + \left(1 + \left\lfloor \frac{t + J_i^1}{T_i} \right\rfloor\right) \cdot C_i^{slow}$$

$$+ \sum_{j\in hp_t(i)} \left(1 + \left\lfloor \frac{max(0; min(W_i^q(t) - M_j^{1,q}; t + D_i^1 - D_j^1)) + J_j^1}{T_j} \right\rfloor\right) \cdot C_j^{slow} + \mathcal{A}_i(t),$$

where $\mathcal{A}_i(t) = \sum^{q}_{\substack{h=1\\ h\neq slow}} C^h_{max_{i,t}} - C_i^q + H_i^{1,q}(t) + (q-1)\cdot L_{max}.$

Proof: To determine the latest starting time of packet m, we identify the busy periods of level $P_{i,t+D_i^1}$ that affect the delay of m. For this, we consider the busy period of level $P_{i,t+D_i^1}$, denoted $bp_{i,t}^q$, in which m is processed on node q and we define $f(q)$ as the first packet processed in $bp_{i,t}^q$ with a priority greater than or equal to $P_{i,t+D_i^1}$. Due to the non-preemptive effect, the execution of $f(q)$ can be delayed once by a packet with a priority less than $P_{i,t+D_i^1}$. Hence, any packet between $f(q)$ and m has a priority greater than or equal to $P_{i,t+D_i^1}$.

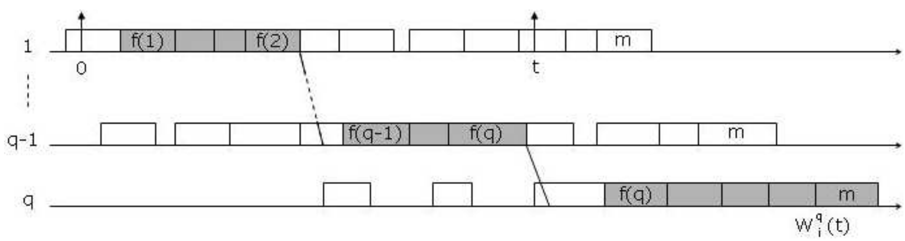

Figure 1: Starting time of packet m in node q

The packet $f(q)$ has been processed in a busy period on node $q-1$ at least of level $P_{i,t+D_i^1}$. Let bp_i^{q-1} be this busy period. We then define $f(q-1)$ as the first packet processed in bp_i^{q-1} with a priority $\geq P_{i,t+D_i^1}$. And so on until the busy period of node 1 in which the packet $f(1)$ is processed (see Figure 1).
For the sake of simplicity, we number consecutively the packets of the considered busy periods of level $P_{i,t+D_i^1}$. Hence, we denote $m'-1$ (respectively $m'+1$) the packet preceding (respectively succeeding) m'. Moreover, in the following, we consider the arrival time of packet $f(1)$ in node 1 as the time origin. By adding parts of the busy periods considered, we can express the starting time of packet m in node q, that is:

the processing time on node 1 of packets $f(1)$ to $f(2)$ $+$ $L_{f(2)}^{1,2}$

$+$ the processing time on node 2 of packets $f(2)$ to $f(3)$ $+$ $L_{f(3)}^{2,3}$

$+$ ...

$+$ the processing time on node q of packets $f(q)$ to $(m-1)$.

$+$ $H_i^{1,q}(t)$, the maximum delay directly due to packets $\in lp(i) \cup \overline{hp}_t(i)$ while visiting nodes 1 to q.

By convention, $f(q+1) = m$. Then, the latest starting time, in node q, of packet m requested at time t on node 1 meets: $W_i^q(t) = \sum_{h=1}^{q}\left(\sum_{g=f(h)}^{f(h+1)} C_{\tau(g)}^h\right) - C_i^q + H_i^{1,q}(t) + (q-1)\cdot L_{max}$. By definition, on any node h, packets $f(h)$ to $f(h+1)$ have a priority higher than or equal to $P_{i,t+D_i^1}$ since all the busy periods we consider are at least of level $P_{i,t+D_i^1}$. We now consider the term $\sum_{h=1}^{q}(\sum_{g=f(h)}^{f(h+1)} C_{\tau(g)}^h)$ and distinguish the nodes visited by the flows before node *slow* and those visited after. Thus, this term is less than or equal to:

$$\underbrace{\sum_{h=1}^{slow-1}\left(\sum_{g=f(h)}^{f(h+1)-1} C_{\tau(g)}^h + C_{\tau(f(h+1))}^h\right)}_{\textit{nodes visited before slow}} + \underbrace{\sum_{g=f(slow)}^{f(slow+1)} C_{\tau(g)}^{slow}}_{\textit{node slow}} + \underbrace{\sum_{h=slow+1}^{q}\left(\sum_{g=f(h)+1}^{f(h+1)} C_{\tau(g)}^h + C_{\tau(f(h))}^h\right)}_{\textit{nodes visited after slow}}.$$

For any node $h \in [1,q]$, for any packet m' visiting h, the processing time of m' on node h is less than $C_{\tau(m')}^{slow}$. Then, as packets are numbered consecutively from $f(1)$ to $f(q+1) = m$, we get inequation (1). By considering that on any node h, the processing time of a packet with a priority greater than or equal to $P_{i,t+D_i^1}$ is less than or equal to $C_{max_{i,t}}^h = \max_{j\in gp(i)\cup hp_t(i)\cup\{i\}}\{C_j^h\}$, we get inequation (2).

$$\sum_{h=1}^{slow-1}\left(\sum_{g=f(h)}^{f(h+1)-1} C_{\tau(g)}^h\right) + \sum_{g=f(slow)}^{f(slow+1)} C_{\tau(g)}^{slow} + \sum_{h=slow+1}^{q}\left(\sum_{g=f(h)+1}^{f(h+1)} C_{\tau(g)}^h\right) - C_i^q \leq \sum_{g=f(1)}^{m} C_{\tau(g)}^{slow} - C_i^q \quad (1)$$

$$\sum_{h=1}^{slow-1} C_{\tau(f(h+1))}^h + \sum_{h=slow+1}^{q} C_{\tau(f(h))}^h \leq \sum_{\substack{h=1\\h\neq slow}}^{q} C_{max_{i,t}}^h \quad (2)$$

By (1) and (2), we get: $\sum_{h=1}^{q}\left(\sum_{g=f(h)}^{f(h+1)} C_{\tau(g)}^h\right) - C_i^q \leq \sum_{g=f(1)}^{m} C_{\tau(g)}^{slow} - C_i^q + \sum_{\substack{h=1\\h\neq slow}}^{q} C_{max_{i,t}}^h$.

The term $\sum_{g=f(1)}^{m} C_{\tau(g)}^{slow}$ corresponds to packets of:

- flow $\tau_j \in gp(i)$ and arrived in node q before m starts its execution;
- flow $\tau_j \in hp_t(i)$ requested at a time $t' \leq t + D_i^1 - D_j^1$ according to Property 10 and arrived in node q before m starts its execution;
- flow τ_i requested at a time less than or equal to t.

The term $\sum_{g=f(1)}^{m} C_{\tau(g)}^{slow}$ is bounded by the maximum workload generated by:

- flows τ_j belonging to $gp(i)$ in the interval $[-J_j^1, \max(0\,;\, W_i^q(t) - M_j^{1,q})]$;
- flows τ_j belonging to $hp_t(i)$ in the interval $[-J_j^1, max(0; min(W_i^q(t) - M_j^{1,q}; t + D_i^1 - D_j^1))]$;
- flow τ_i in the interval $[-J_i^1, t]$.

Indeed, any packet entered the distributed system before $f(1)$ does not interfere with the considered packets processed in the selected busy periods. It is the same for any packet belonging to $gp(i)$ and arriving in node q after m starts its execution. Then, a packet of any flow $\tau_j \in gp(i)$ does not delay m if it is requested in node 1 after the time: $W_i^q(t) - M_j^{1,q}$, where $M_j^{1,q}$ denotes the minimum time taken by a packet of flow τ_j to go from node 1 to node q. In the same way, any packet belonging to $hp_t(i)$ and requested in node 1 after $W_i^q(t) - M_j^{1,q}$ or after $t + D_i^1 - D_j^1$ does not delay m. Thus, we get $\sum_{g=f(1)}^{m} C_{\tau(g)}^{slow}$ less than or equal to:

$$\sum_{j\in gp(i)} \left(1+\left\lfloor \frac{\max\left(0\,;W_i^q(t)-M_j^{1,q}\right)+J_j^1}{T_j}\right\rfloor\right)\cdot C_j^{slow} + \sum_{j\in hp_t(i)} \left(1+\left\lfloor \frac{max(0;min(W_i^q(t)-M_j^{1,q};t+D_i^1-D_j^1))+J_j^1}{T_j}\right\rfloor\right)\cdot C_j^{slow}$$

$$+\left(1+\left\lfloor \frac{t+J_i^1}{T_i}\right\rfloor\right)\cdot C_i^{slow}.$$

As $W_i^q(t) = \sum_{g=f(1)}^{m} C_{\tau(g)}^{slow} + \sum_{\substack{h=1\\ h\neq slow}}^{q} C_{max_{i,t}}^{h} - C_i^q + H_i^{1,q}(t) + (q-1)\cdot L_{max}$, we get Lemma 7. ■

5.3 Analysis of the latest starting time

We now focus on the following series that we denote $\mathcal{W}_i^q(t)$:

$$W_i^{q^{(0)}}(t) = \sum_{j\in gp(i)\cup hp_t(i)} C_j^{slow} + \left(1+\left\lfloor \frac{t+J_i^1}{T_i}\right\rfloor\right)\cdot C_i^{slow} + \mathcal{A}_i(t)$$

$$W_i^{q^{(p+1)}}(t) = \sum_{j\in gp(i)} \left(1+\left\lfloor \frac{\max\left(0\,;W_i^{q^{(p)}}(t)-M_j^{1,q}\right)+J_j^1}{T_j}\right\rfloor\right)\cdot C_j^{slow} + \left(1+\left\lfloor \frac{t+J_i^1}{T_i}\right\rfloor\right)\cdot C_i^{slow}$$
$$+\sum_{j\in hp_t(i)} \left(1+\left\lfloor \frac{max(0;min(W_i^{q^{(p)}}(t)-M_j^{1,q};t+D_i^1-D_j^1))+J_j^1}{T_j}\right\rfloor\right)\cdot C_j^{slow} + \mathcal{A}_i(t),$$

with $\mathcal{A}_i(t) = \sum_{\substack{h=1\\ h\neq slow}}^{q} C_{max_{i,t}}^{h} - C_i^q + H_i^{1,q}(t) + (q-1)\cdot L_{max}$.

We first prove the existence of $W_i^q(t)$, solution of the equation given in Lemma 7. Then, we show that only a limited set of arrival times in the distributed system has to be tested to obtain the latest starting time on node q of a packet of τ_i. We finally show how to compute the worst case end-to-end response time of any flow τ_i.

Condition 2 *For any time flow τ_i, for any time $t \geq -J_i^1$ where a packet of τ_i is requested on node 1, if $U^{slow}_{gp(i)\cup hp_t(i)} < 1$, where $U^{slow}_{gp(i)\cup hp_t(i)}$ denotes the utilization factor on node slow for the flows belonging to $gp(i) \cup hp_t(i)$, then $\mathcal{W}_i^q(t)$ is convergent.*

Proof: For any time $t \geq -J_i^1$, the series $\mathcal{W}_i(t)$ is a non-decreasing series as the floor function is non-decreasing. Moreover, this series is upper bounded by: $X^{slow}/(1 - U^{slow}_{gp(i)\cup hp_t(i)})$, where:

$$X^{slow} \;=\; \sum_{j\in gp(i)\cup hp_t(i)} \left(1+\frac{J_j^1}{T_j}\right)\cdot C_j^{slow} + \left(1 + \left\lfloor \frac{t+J_i^1}{T_i}\right\rfloor\right)\cdot C_i^{slow} + \mathcal{A}_i(t).$$

Indeed, by recurrence, we have: $W_i^{q^{(0)}}(t) \leq X^{slow}$, that is less than or equal to: $X^{slow}/(1 - U^{slow}_{gp(i)\cup hp_t(i)})$, assuming $U^{slow}_{gp(i)\cup hp_t(i)} < 1$. We now assume that the recurrence is true at rank p and show that it is true at rank $p+1$. As $min(W_i^{q^{(p)}}(t) - M_j^{1,q}; t + D_i^1 - D_j^1) \leq W_i^{q^{(p)}}(t) - M_j^{1,q}$, we have $W_i^{q^{(p+1)}}(t)$ less than or equal to:

$$W_i^{q^{(p)}}(t)\cdot \sum_{j\in gp(i)\cup hp_t(i)} \frac{C_j^{slow}}{T_j} + \sum_{j\in gp(i)\cup hp_t(i)} \left(1+\frac{J_j^1}{T_j}\right)\cdot C_j^{slow} + \left(1+\left\lfloor\frac{t+J_i^1}{T_i}\right\rfloor\right)\cdot C_i^{slow} + \mathcal{A}_i(t)$$

$$\leq \quad W_i^{q^{(p)}}(t)\cdot U^{slow}_{gp(i)\cup hp_t(i)} + X^{slow} \quad \leq \quad X^{slow}\cdot U^{slow}_{gp(i)\cup hp_t(i)}/(1 - U^{slow}_{gp(i)\cup hp_t(i)}) + X^{slow}$$

$$\leq \quad X^{slow}/(1 - U^{slow}_{gp(i)\cup hp_t(i)}).$$

The series $\mathcal{W}_i^q(t)$ is non-decreasing and upper bounded. Hence, this series is convergent. ■

We recall that a necessary condition for the feasibility of a set of flows is: $\forall\, h \in \mathcal{L}$, $U^h \leq 1$, where $U^h = \sum_{j=1}^n C_j^h/T_j$ denotes the utilization factor on node h. Hence, Lemma 2 is not restrictive. Indeed, if $U^{slow} \leq 1$, then $U^{slow}_{hp_t(i)} < 1$.

We now establish the three following lemmas that enable to limit the set of arrival times in the distributed system when computing the worst case end-to-end response time.

Lemma 8 *Let $\mathcal{S}$ be the ordered set of times $k \cdot T_j - J_j^1 - D_i^1 + D_j^1 \geq -J_i^1$, where $k \in \mathbb{N}$ and $j \in sp(i) \cup \{i\}$. Let t_1 and t_2 be two consecutive times of $\mathcal{S}$. Then, $\forall\, t' \in [t_1, t_2)$, $W_i^q(t') = W_i^q(t_1)$.*

Proof: We first show that $hp_{t_1}(i) = hp_{t'}(i)$. For any $\tau_j \in hp_{t_1}(i)$, we have: $D_j^1 - J_j^1 \leq t_1 + D_i^1$. As $t_1 \leq t'$, we get: $D_j^1 - J_j^1 \leq t' + D_i^1$. Hence τ_j belongs to $hp_{t'}(i)$, and $hp_{t_1}(i) \subset hp_{t'}(i)$.
We now assume that there exists $\tau_j \in hp_{t'}(i)$ that does not belong to $hp_{t_1}(i)$. Hence, we get: $t' \geq D_j^1 - J_j^1 - D_i^1$ and $t_1 < D_j^1 - J_j^1 - D_i^1$. It follows that there exists an element of the ordered set (S) in (t_1, t'): a contradiction with our assumption. We then have $hp_{t_1}(i) = hp_{t'}(i)$. We deduce $\overline{hp}_{t_1}(i) = \overline{hp}_{t'}(i)$ and $\mathcal{A}_i(t_1) = \mathcal{A}_i(t')$.
We then consider the series $\mathcal{W}_i^q(t)$ and prove the lemma by recurrence. By assumption, no packet of τ_i can be requested before time $-J_i^1$. Hence, we only consider times $t \geq -J_i^1$. By

definition, if t_1 and t_2 are two consecutive times of set $\mathcal{S}$, then we get: $\forall\, j \in hp_t(i) \cup \{i\}$, $\forall\, t' \in [t_1, t_2)$, $\lfloor (t' + D_i^1 - D_j^1 + J_j^1)/T_j \rfloor = \lfloor (t_1 + D_i^1 - D_j^1 + J_j^1)/T_j \rfloor$ and $\lfloor (t' + J_i^1)/T_i \rfloor = \lfloor (t_1 + J_i^1)/T_i \rfloor$. Therefore, the lemma is met at rank 0. Indeed, $W_i^{q^{(0)}}(t') = W_i^{q^{(0)}}(t_1)$.
Assuming that the recurrence is true at rank p, we show that it is true at rank $p+1$. $W_i^{q^{(p+1)}}(t')$ is equal to:

$$\sum_{j \in gp(i)} \left(1 + \left\lfloor \frac{max(0; W_i^{q^{(p)}}(t') - M_j^{1,q}) + J_j^1}{T_j} \right\rfloor\right) \cdot C_j^{slow} + \left(1 + \left\lfloor \frac{t' + J_i^1}{T_i} \right\rfloor\right) \cdot C_i^{slow}$$

$$+ \sum_{j \in hp_{t'}(i)} \left(1 + \left\lfloor \frac{max(0; min(W_i^{q^{(p)}}(t') - M_j^{1,q}; t' + D_i^1 - D_j^1)) + J_j^1}{T_j} \right\rfloor\right) \cdot C_j^{slow} + \mathcal{A}_i(t')$$

$$= \sum_{j \in gp(i)} \left(1 + \left\lfloor \frac{max(0; W_i^{q^{(p)}}(t_1) - M_j^{1,q}) + J_j^1}{T_j} \right\rfloor\right) \cdot C_j^{slow} + \left(1 + \left\lfloor \frac{t_1 + J_i^1}{T_i} \right\rfloor\right) \cdot C_i^{slow}$$

$$+ \sum_{j \in hp_{t_1}(i)} \left(1 + \left\lfloor \frac{max(0; min(W_i^{q^{(p)}}(t_1) - M_j^{1,q}; t_1 + D_i^1 - D_j^1)) + J_j^1}{T_j} \right\rfloor\right) \cdot C_j^{slow} + \mathcal{A}_i(t_1)$$

$$= W_i^{q^{(p+1)}}(t_1).$$

■

Lemma 9 *For any flow* $\tau_i \in \tau$*, for any* t *greater than or equal to* $\max_{k \in sp(i)}\{D_k^1\} - D_i^1 - \min_{k \in sp(i)}\{J_k^1\}$*, we have:* $\overline{hp}_t(i) = \emptyset$.

Proof: For any flow $\tau_i \in \tau$, for any $j \in sp(i)$, we have $D_j^1 - J_j^1 \leq \max_{k \in sp(i)}\{D_k^1\} - \min_{k \in sp(i)}\{J_k^1\}$. Hence for any time $t \geq \max_{k \in sp(i)}\{D_k^1\} - D_i^1 - \min_{k \in sp(i)}\{J_k^1\}$, we get $D_j^1 - J_j^1 \leq t + D_i^1$. Hence, $j \in hp_t(i)$. It follows $\overline{hp}_t(i) = \emptyset$. ■

Lemma 10 *For any flow* $\tau_i \in \tau$*, for any time* t *greater than or equal to* $\max_{k \in sp(i)}\{D_k^1\} - D_i^1 - \min_{k \in sp(i)}\{J_k^1\}$*, we have:* $W_i^q(t + B_i^{slow}) \leq W_i^q(t) + B_i^{slow}$*, with:* $B_i^{slow} = \sum_{j \in gp(i) \cup sp(i) \cup \{i\}} \lceil B_i^{slow}/T_j \rceil \cdot C_j^{slow}$.

Proof: For any flow $\tau_i \in \tau$, for any time $t \geq -J_i$, we have $hp_{t+B_i^{slow}}(i) \supseteq hp_t(i)$. We can then write $hp_{t+B_i^{slow}}(i) = hp_t(i) \cup (\overline{hp}_t(i) \cap hp_{t+B_i^{slow}}(i))$. According to the previous lemma, for any time $t \geq \max_{k \in sp(i)}\{D_k^1\} - D_i^1 - \min_{k \in sp(i)}\{J_k^1\}$, $\overline{hp}_t(i) = \emptyset$, we then get $hp_{t+B_i^{slow}}(i) = hp_t(i)$.
Moreover, as $\overline{hp}_{t+B_i^{slow}}(i) \subseteq \overline{hp}_t(i)$, we get for any time $t \geq \max_{k \in sp(i)}\{D_k^1\} - D_i^1 - \min_{k \in sp(i)}\{J_k^1\}$, $\overline{hp}_{t+B_i^{slow}}(i) = \overline{hp}_t(i)$.

Hence $H_i^{1,q}(t+B_i^{slow}) = H_i^{1,q}(t)$= the delay introduced by flows τ_j, with $j \in lp(i)$.

We consider the series $\mathcal{W}_i^q(t+B_i^{slow})$ and prove the lemma by recurrence, for any time $t \geq \max_{k\in sp(i)}\{D_k^1\} - D_i^1 - \min_{k\in sp(i)}\{J_k^1\}$.
At rank 0, we have:

$W_i^{q^{(0)}}(t+B_i^{slow}) = \sum_{j\in gp(i)\cup hp_{t+B_i^{slow}}(i)} C_j^{slow} + \left(1 + \lfloor\frac{t+B_i^{slow}+J_i^1}{T_i}\rfloor\right)\cdot C_i^{slow} + \mathcal{A}_i(t+B_i^{slow})$.

As $hp_{t+B_i^{slow}}(i) = hp_t(i)$ and $\lfloor a+b \rfloor \leq \lfloor a \rfloor + \lceil b \rceil$, we have:
$W_i^{q^{(0)}}(t+B_i^{slow}) \leq \sum_{j\in gp(i)\cup hp_t(i)} C_j^{slow} + \left(1 + \lfloor\frac{t+J_i^1}{T_i}\rfloor\right)\cdot C_i^{slow} + \lceil\frac{B_i^{slow}}{T_i}\rceil\cdot C_i^{slow} + \mathcal{A}_i(t+B_i^{slow})$.
As $hp_{t+B_i^{slow}}(i) = hp_t(i)$ and $\overline{hp}_t(i) = \overline{hp}_{t+B_i^{slow}}(i) = \emptyset$, we have $\mathcal{A}_i(t+B_i^{slow}) = \mathcal{A}_i(t)$.

We finally get:

$$W_i^{q^{(0)}}(t+B_i^{slow}) \leq W_i^{q^{(0)}}(t) + \lceil\frac{B_i^{slow}}{T_i}\rceil\cdot C_i^{slow} \leq W_i^{q^{(0)}}(t) + B_i^{slow}.$$

Assuming that this property is true at rank p, we prove it at rank $p+1$.
$W_i^{q^{(p+1)}}(t+B_i^{slow})$ is equal to:

$$\sum_{j\in gp(i)}\left(1+\left\lfloor\frac{\max\left(0\,;W_i^{q^{(p)}}(t+B_i^{slow})-M_j^{1,q}\right)+J_j^1}{T_j}\right\rfloor\right)\cdot C_j^{slow} + \left(1+\left\lfloor\frac{t+B_i^{slow}+J_i^1}{T_i}\right\rfloor\right)\cdot C_i^{slow}$$
$$+\sum_{j\in hp_{t+B_i^{slow}}(i)}\left(1+\left\lfloor\frac{max(0;min(W_i^{q^{(p)}}(t+B_i^{slow})-M_j^{1,q};t+B_i^{slow}+D_i^1-D_j^1))+J_j^1}{T_j}\right\rfloor\right)\cdot C_j^{slow} + \mathcal{A}_i(t+B_i^{slow}).$$

As $W_i^{q^{(p)}}(t+B_i^{slow}) \leq W_i^{q^{(p)}}(t) + B_i^{slow}$, $\lfloor a+b \rfloor \leq \lfloor a \rfloor + \lceil b \rceil$ and $\forall a \geq 0$, $max(0;a+b) \leq a + max(0;b)$, we get $W_i^{q^{(p+1)}}(t+B_i^{slow})$ bounded by:

$$\sum_{j\in gp(i)}\left(1+\left\lfloor\frac{\max\left(0\,;W_i^{q^{(p)}}(t)-M_j^{1,q}\right)+J_j^1}{T_j}\right\rfloor + \left\lceil\frac{B_i^{slow}}{T_j}\right\rceil\right)\cdot C_j^{slow} + \left(1+\left\lfloor\frac{t+J_i^1}{T_i}\right\rfloor + \left\lceil\frac{B_i^{slow}}{T_i}\right\rceil\right)\cdot C_i^{slow}$$
$$+\sum_{j\in hp_{t+B_i^{slow}}(i)}\left(1+\left\lfloor\frac{max(0;min(W_i^{q^{(p)}}(t)-M_j^{1,q};t+D_i^1-D_j^1))+J_j^1}{T_j}\right\rfloor + \left\lceil\frac{B_i^{slow}}{T_j}\right\rceil\right)\cdot C_j^{slow} + \mathcal{A}_i(t+B_i^{slow})$$

As by definition, we have: $B_i^{slow} = \sum_{j\in gp(i)\cup sp(i)\cup\{i\}} \lceil B_i^{slow}/T_j\rceil \cdot C_j^{slow}$, we get:
$W_i^{q^{(p+1)}}(t+B_i^{slow})$ upper bounded by:

$$B_i^{slow} + \sum_{j \in gp(i)} \left(1 + \left\lfloor \frac{\max\left(0\,; W_i^{q^{(p)}}(t) - M_j^{1,q}\right) + J_j^1}{T_j} \right\rfloor \right) \cdot C_j^{slow} + \left(1 + \left\lfloor \frac{t+J_i^1}{T_i} \right\rfloor\right) \cdot C_i^{slow}$$
$$+ \sum_{j \in hp_{t+B_i^{slow}}(i)} \left(1 + \left\lfloor \frac{max(0; min(W_i^{q^{(p)}}(t) - M_j^{1,q}; t + D_i^1 - D_j^1)) + J_j^1}{T_j} \right\rfloor \right) \cdot C_j^{slow} + \mathcal{A}_i(t + B_i^{slow}).$$

As for any time $t \geq \max_{k \in sp(i)}\{D_k^1\} - D_i^1 - \min_{k \in sp(i)}\{J_k^1\}$,
$hp_{t+B_i^{slow}}(i) = hp_t(i)$, $\overline{hp}_{t+B_i^{slow}}(i) = \overline{hp}_t(i)$ and $H_i^{1,q}(t + B_i^{slow}) = H_i^{1,q}(t)$,
we finally get $W_i^{q^{(p+1)}}(t + B_i^{slow}) \leq B_i^{slow} + W_i^{q^{(p+1)}}(t)$. ■

5.4 Worst case end-to-end response time

The worst case end-to-end response time of any packet m belonging to flow τ_i is equal to the latest starting time of m in node q, plus C_i^q, minus t, the arrival time of packet m in the distributed system. More precisely, it is equal to: $W_i^q(t) + C_i^q - t$. The worst case end-to-end response time of flow τ_i is equal to the maximum of the worst case end-to-end response times of its packets. Moreover, to compute this worst case response time, we have only to consider times $t \geq -J_i^1$ such that: $t = k \cdot T_j - J_j^1 - D_i^1 + D_j^1 < \max_{k \in sp(i)}\{D_k^1\} - D_i^1 - \min_{k \in sp(i)}\{J_k^1\} + B_i^{slow}$, with $k \in \mathbb{N}$ and $j \in sp(i) \cup \{i\}$ (see Lemmas 8 and 10).

Property 11 *When all flows follow the same line $\mathcal{L}$ consisting of q nodes numbered from* 1 *to q and are scheduled according to* FP/EDF*, the worst case end-to-end response time of any flow τ_i meets:*

$$\begin{aligned}
Rmax_i^{\mathcal{L}} = \quad & \max_{t \in \mathcal{S}'} \{W_i^q(t) + C_i^q - t\}, \textit{ where:} \\
W_i^q(t) = \quad & \sum_{j \in gp(i)} \left(1 + \left\lfloor \frac{\max\left(0\,; W_i^q(t) - M_j^{1,q}\right) + J_j^1}{T_j} \right\rfloor \right) \cdot C_j^{slow} + \left(1 + \left\lfloor \frac{t+J_i^1}{T_i} \right\rfloor\right) \cdot C_i^{slow} \\
& + \sum_{j \in hp_t(i)} \left(1 + \left\lfloor \frac{\max\left(0\,; min(W_i^q(t) - M_j^{1,q}; t + D_i^1 - D_j^1)\right) + J_j^1}{T_j} \right\rfloor \right) \cdot C_j^{slow} + \mathcal{A}_i(t), \\
\mathcal{A}_i(t) = \quad & \sum_{\substack{h=1 \\ h \neq slow}}^{q} C_{max_{i,t}}^h - C_i^q + H_i^{1,q}(t) + (q-1) \cdot Lmax
\end{aligned}$$

and $\mathcal{S}'$ denotes the ordered set of times $t \geq -J_i^1$ such that: $t = k \cdot T_j - J_j^1 - D_i^1 + D_j^1 <$ $\max_{k \in sp(i)}\{D_k^1\} - D_i^1 - \min_{k \in sp(i)}\{J_k^1\} + B_i^{slow}$, with $k \in \mathbb{N}$ and $j \in sp(i) \cup \{i\}$.

Proof: This property can be deduced from Lemmas 7, 8 and 10. ■

It is important to notice that the cardinal of set S' only depends on flow τ_i and on flows having the same priority. We can also notice that in the single node case, this bound is exact.

Property 12 *In the single node case, the bound given by property 11 is this given in the uniprocessor context. Indeed, we get:*

$$R_{max_i} = \max_{t \in \mathcal{S}'} \left\{ W_i^1(t) + C_i^1 - t \right\}, \text{ where:}$$

$$W_i^1(t) = \sum_{j \in gp(i)} \left(1 + \left\lfloor \frac{W_i^1(t)+J_j^1}{T_j} \right\rfloor\right) \cdot C_j^1 + \sum_{j \in hp_t(i)} \left(1 + \left\lfloor \frac{min(W_i^1(t); t+D_i^1-D_j^1)+J_j^1}{T_j} \right\rfloor\right) \cdot C_j^1$$
$$+ \left\lfloor \frac{t+J_i^1}{T_i} \right\rfloor \cdot C_i^1 + \max(0\,; C^1_{\overline{max}_{i,t}} - 1),$$

and $\mathcal{S}'$ *denotes the ordered set of times* $t \geq -J_i^1$ *such that:* $t = k \cdot T_j - J_j^1 - D_i^1 + D_j^1 < \max_{k \in sp(i)}\{D_k^1\} - D_i^1 - \min_{k \in sp(i)}\{J_k^1\} + B_i^{slow}$*, with* $k \in \mathbb{N}$ *and* $j \in sp(i) \cup \{i\}$*.*

5.5 Evaluation of the delay due to packets belonging to $lp(i) \cup \overline{hp}_t(i)$

We improve in this section the bound given in Lemma 6 on the maximum delay incurred by a packet of τ_i, requested at time t on node 1, directly due to packets belonging to $lp(i) \cup \overline{hp}_t(i)$ in the particular case when network delay is constant and the processing time on node h is equal to C^h for any flow. This case is frequently encountered as along a path, all the packets have the same size, equal to the minimum *Maximum Transmission Unit* (MTU) on the path, leading to avoid the segmentation and reassembling on the intermediate nodes. This alleviates the processing on any intermediate node.

For instance, in IPv6, the Path MTU Discovery determines this minimum MTU. In the case of a single line, this leads to consider a processing time equal to C^h for any flow on any node h. Moreover, point-to-point links are an example of links providing constant network delays.

Lemma 11 *On any node* $h \in [1, q]$*, the processing time of any flow being equal to* C^h *and the network delay being constant, the interarrival time of two successive packets on any node* $h \in (1, q]$ *is at least equal to* $\max_{k=1..h-1}\{C^k\}$*.*

Proof: As the network delay is constant, the interarrival time of two successive packets on any node $h \in (1, q]$ is equal to their interdeparture time on node $h-1$. Let k be the slowest node among nodes 1 to $h-1$. As the processing delay on any node is constant, the interdeparture time on node k is at least equal to C^k. As on any node $k+1$ to $h-1$, the processing time is less than or equal to C^k, the interarrival time on node h is at least equal to C^k. ■

Property 13 *When, on any node* $h \in [1, q]$*, the processing time of all flows is equal to* C^h *and the network delay is constant, then for the packet of any flow* τ_i*, requested at time t on node* 1*, the maximum delay directly due to packets belonging to* $lp(i) \cup \overline{hp}_t(i)$ *while visiting nodes* 1 *to* $h+1$*,* $h \in [1, q)$*, meets:*

$$\begin{cases} H_i^{1,h+1}(t) = H_i^{1,h}(t) \quad if \quad C^{h+1} \leq \max_{k=1..h}\{C^k\} \\ H_i^{1,h+1}(t) \leq H_i^{1,h}(t) + \max\left(0; C^{h+1}_{max_{i,t}} - 1\right) \quad otherwise, \end{cases}$$

where $H_i^{1,1}(t) = \max\left(0; C^1_{max_{i,t}} - 1\right)$.

Proof: We assume that on any node $h \in [1, q]$, the processing time of all flows is equal to C^h and the network delay is constant. We consider a packet m of flow τ_i, requested at time t on node 1 and distinguish two cases:

- $C^{h+1} \leq \max_{k=1..h} C^k$. By applying Lemma 11, no additional delay due to packets belonging to $\overline{hp}_t(i)$ is incurred by τ_i. Hence, $H_i^{1,h+1}(t) = H_i^{1,h}(t)$.
- $C^{h+1} > \max_{k=1..h} C^k$. By applying Lemma 11, an additional delay due to packets belonging to $\overline{hp}_t(i)$ is incurred by τ_i on node $h+1$. Hence, we have: $H_i^{1,h+1}(t) \leq H_i^{1,h}(t) + \max\left(0; C^{h+1}_{max_{i,t}} - 1\right)$. ■

6 Comparative evaluation in the distributed case

We first recall the computation principle of the worst case end-to-end response time in the distributed case when applying the holistic approach. Then, we give several examples that illustrate how close our results are to the exact results. We also compare our results to these obtained by the holistic approach.

6.1 Worst case end-to-end response time by the holistic approach

We now apply the holistic approach to compute the worst case end-to-end response time of any flow τ_i, when all flows follow the same line $\mathcal{L}$. We denote $Rmax_j^h$ (respectively $Rmin_j^h$) the maximum (respectively the minimum) response time experienced by packets of any flow τ_j in node h and J_j^h its worst case jitter when entering node h.

The holistic approach proceeds iteratively and starts with node 1. Knowing the value of J_j^1 for any $j \in [1, n]$, we compute $Rmax_j^1$ using Property 12 and $Rmin_j^1 = C_j^1$, $\forall j \in [1, n]$. We proceed in the same way for any node h, $h \in (1, q]$.

Knowing the value of $J_j^h = \sum_{k=1..h-1}(Rmax_j^k - Rmin_j^k) + (h-1) \cdot (Lmax - Lmin)$, $\forall j \in [1,n]$, we compute $Rmax_j^h$ using Property 12 and $Rmin_j^h = C_j^h$.

A bound on the end-to-end response time of flow τ_i is given by: $\sum_{h=1}^{q} Rmax_i^h - \sum_{h=2}^{q} J_i^h + (q-1) \cdot Lmax$.

6.2 Examples

In this section, we give examples of bounds on the end-to-end response times of flows in a distributed system, when all flows follow the same line consisting of 5 nodes. We assume that $\tau = \{\tau_1, \tau_2, \tau_3, \tau_4, \tau_5\}$, all the flows having a period equal to 36 and entering the distributed system without jitter. For any flow τ_i, we have $D_i^1 = \lfloor D_i/q \rfloor$. The load is equal to 83,33%. Moreover, there is only one flow in the highest priority 3, whereas two flows share priority 2 and priority 1. Finally, we have $Lmax = Lmin = 1$.

As part of this paper, a simulation tool has been developed for providing the exhaustive solution of a real-time scheduling problem in a network. Indeed, once the different parameters have been specified, all the possible scenarios are generated and feasibility of the flow set is checked for each of them. The simulation result is a file containing the exact worst case end-to-end response time of each flow.

We consider four configurations: the processing times of the packets (i) decrease node after node, (ii) increase node after node, (iii) are not ordered and (iv) are the same in any node visited. The following tables give for each configuration and for any flow τ_i, $i \in [1,5]$, the exact value of its worst case end-to-end response time and the value computed according to the trajectory approach. To show the improvement of our results compared with those obtained by the classical technique, we also include in these tables the value computed according to the holistic approach.

We observe that the values provided by the trajectory approach are exact for all flows in configurations (i) and (iv). In configurations (ii) and (iii), the values provided by the trajectory approach are exact only for the lowest priority flows. This can be explained by the overestimation of $H_i^{1,q}(t)$, the delay directly due to flows belonging to $lp(i) \cup \overline{hp}_t(i)$, in such configurations.

The bounds provided by the holistic approach are very pessimistic for small priority flows. For instance, in configuration (iv), the value provided by the holistic approach is 5.5 times the exact one. Even for the highest priority flow, the bound can be pessimistic, as we can see in (iv), where it is equal to 1.6 time the exact value.

Table 4: Worst case end-to-end response time of any flow τ_i, $i \in [1,5]$

(i) Processing times decrease node after node
$C_i^1 = 6$, $C_i^2 = 5$, $C_i^3 = 4$, $C_i^4 = 3$, $C_i^5 = 2$

Flow	P_i	D_i	Exact	Trajectory	Holistic
τ_1	1	47	47	47	178
τ_2	1	50	48	48	183
τ_3	2	44	40	40	78
τ_4	2	45	41	41	83
τ_5	3	39	29	29	39

(ii) Processing times increase node after node
$C_i^1 = 2$, $C_i^2 = 3$, $C_i^3 = 4$, $C_i^4 = 5$, $C_i^5 = 6$

Flow	P_i	D_i	Exact	Trajectory	Holistic
τ_1	1	47	47	47	191
τ_2	1	50	48	48	196
τ_3	2	44	44	50	80
τ_4	2	45	45	51	85
τ_5	3	39	38	39	39

(iii) Processing times are not ordered
$C_i^1 = 3$, $C_i^2 = 5$, $C_i^3 = 2$, $C_i^4 = 6$, $C_i^5 = 4$

Flow	P_i	D_i	Exact	Trajectory	Holistic
τ_1	1	47	47	47	203
τ_2	1	50	48	48	208
τ_3	2	44	43	46	82
τ_4	2	45	44	47	87
τ_5	3	39	34	35	39

(iv) Processing times are the same in any node
$C_i^1 = 4$, $C_i^2 = 4$, $C_i^3 = 4$, $C_i^4 = 4$, $C_i^5 = 4$

Flow	P_i	D_i	Exact	Trajectory	Holistic
τ_1	1	47	39	39	221
τ_2	1	50	40	40	226
τ_3	2	44	34	34	90
τ_4	2	45	35	35	95
τ_5	3	39	27	27	43

7 Conclusion

In this paper, we have established new results for Fixed Priority scheduling in the uniprocessor and distributed cases. By assuming that flows sharing the same priority are scheduled EDF, we have revisited classical results in a uniprocessor context. As our solution enables to improve the worst case response times, any set of flows feasible with the classical solution is feasible with ours. The converse is false, as shown by an example. Moreover, we have shown that any sporadic set such that all flows sharing the same priority have the same processing time, feasible with FP/FIFO is feasible with FP/EDF. Then, we have established new results in a distributed context, assuming that all flows follow the same sequence of nodes. With our solution, the computation of the absolute deadline of a packet is made only once on the first visited node. Packets are scheduled FP/EDF according to their absolute deadlines and their static priorities, that are unchanged on each visited node. This solution ensures that if a packet m has a generalized priority higher than a packet m' on a node, this is true on any visited node. We have then shown how to compute an upper bound on the end-to-end response time of any flow with a worst case analysis using the trajectory approach. Thus, we have determined the worst case end-to-end response time of any flow and we have compared these results with the exact values and those provided by the classical holistic approach. We have shown that the bound given by the trajectory approach is reached in various configurations, whereas the holistic approach provides only a bound that can be very pessimistic.

References

[1] K. Tindell, A. Burns, A. J. Wellings, *Analysis of hard real-time communications*, Real-Time Systems, Vol. 9, pp. 147-171, 1995.

[2] J. Liu, *Real-time systems*, Prentice Hall, New Jersey, 2000.

[3] K. Jeffay, D. F. Stanat, C. U. Martel, *On non-preemtive scheduling of periodic and sporadic tasks*, IEEE Real-Time Systems symposium, pp. 129-139, San Antonio, USA, December 1991.

[4] L. George, N. Rivierre, M. Spuri, *Preemptive and non-preemptive scheduling real-time uniprocessor scheduling*, INRIA Research Report No 2966, September 1996.

[5] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, *An architecture for Differentiated Services*, RFC 2475, December 1998.

[6] R. Braden, D. Clark, S. Shenker, *Integrated services in the Internet architecture: an overview*, RFC 1633, June 1994.

[7] S. Baruah, R. Howell, L. Rosier, *Algorithms and complexity concerning the preemptive scheduling of periodic real-time tasks on one processor*, Real-Time Systems, 2, p 301-324, 1990.

[8] V. Sivaraman, F. Chiussi, M. Gerla, *End-to-end statistical delay service under GPS and EDF scheduling: a comparaison study*, INFOCOM'2001, Anchorage, April 2001.

[9] M. Vojnovic, J. Le Boudec, *Stochastic analysis of some expedited forwarding networks*, INFOCOM'2002, New York, June 2002.

[10] F. Chiussi, V. Sivaraman, *Achieving high utilization in guaranteed services networks using early-deadline-first scheduling*, IWQoS'98, Napo, California, May 1998.

[11] L. George, D. Marinca, P. Minet, *A solution for a deterministic QoS in multimedia systems*, International Journal on Computer and Information Science, Vol.1, N3, September 2001.

[12] K. Tindell, J. Clark, *Holistic schedulability analysis for distributed hard real-time systems*, Microprocessors and Microprogramming, Euromicro Journal, Vol. 40, 1994.

[13] J. Le Boudec, P. Thiran, *Network calculus: A theory of deterministic queuing systems for the Internet*, Springer Verlag, LNCS 2050, September 2003.

[14] A. Parekh, R. Gallager, *A generalized processor sharing approach to flow control in integrated services networks: the multiple node case*, IEEE ACM Transactions on Networking, Vol.2, N2, 1994.

[15] L. Georgiadis, R. Guérin, V. Peris, K. Sivarajan, *Efficient network QoS provisioning based on per node traffic shaping*, IEEE/ACM Transactions on Networking, Vol. 4, No. 4, August 1996.

[16] V. Sivaraman, F. Chiussi, M. Gerla, *Traffic shaping for end-to-end delay guarantees with EDF scheduling*, IWQoS'2000, Pittsburgh, June 2000.

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399